C02047: Doctor of Philosophy
CRICOS Code: 058666A
33875 PhD Thesis: Computer Systems
March 2021

A Study on

Feature Analysis and Ensemble-based Intrusion Detection Scheme using CICIDS-2017 dataset

Upasana Nagar

School of Electrical and Data Engineering
Faculty of Engineering & IT
University of Technology Sydney
NSW - 2007, Australia

A Study on

Feature Analysis and Ensemble-based Intrusion Detection Scheme using CICIDS-2017 dataset

A thesis submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Computer Systems

by

Upasana Nagar

under the supervison of

Dr. Priyadarsi Nanda

to

School of Electrical and Data Engineering
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

March 2021

Abstract

One of the primary security research challenges faced by traditional IDS methods is their inability to handle large volumes of network data and detect modern cyber-attacks with high detection accuracy and low false alarms. Hence, there is a need for efficient and reliable IDS schemes that can tackle this ever-changing cybersecurity paradigm. Machine learning techniques are hence, becoming very popular in designing modern intrusion detection systems. Several supervised and unsupervised machine learning techniques have been used in literature; however, the IDS classification efficiency is affected by noisy data in high dimensional datasets. The role of feature selection is significant as the feature selection process eliminates the redundant and noisy data and further selecting optimal feature subset enables reduction of high dimensional IDS datasets. Machine learning algorithms are extensively being used for intrusion detection. However, research has proved that the performance of multiple classifier-based IDS is far better than an IDS classifier, which has given us the motivation to develop an ensemble-based intrusion detection model. Lastly, the benchmark IDS datasets currently being used for the evaluation of IDS schemes are outdated and do not represent modern-day attacks. The CICIDS - 2017 dataset is offered by the University of New Brunswick. It is the latest publicly available dataset for intrusion detection. However, there are a significantly low number of research studies conducted using this dataset which also focus on optimal feature selection. This dataset has a good potential to be used as a future benchmark intrusion detection dataset as it covers the modern-day system setup and threat profile and the dependency on outdated IDS datasets can be removed. There is a need to benchmark the performance of modern IDS datasets using machine learning ensemble-based classifiers. This thesis aims to address the issues by proposing a new intrusion detection framework using ensemble-based feature selection method for generating a low dimensionality feature subset and ensemble-based intrusion detection framework to benchmark the performance of the CICIDS - 2017 dataset. The proposed scheme is beneficial for research community as it combines the use of the latest available IDS dataset with ensemble technique for feature selection and ensemble-based intrusion detection model.

AUTHOR'S DECLARATION

I, *Upasana Nagar* declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *School of Electrical and Data Engineering*, *FEIT* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note:
Signature removed prior to publication.

[Upasana Nagar]

DATE: 30th March, 2021

Dedication

I dedicate this achievement to the Supreme Almighty ...
For bestowing the blessing called "Life" ...
&
Immensely grateful for His guidance to "Turn Inwards" !!!

***

I also dedicate this achievement to my parents ...
Papa & Mumma, you are my friend, philosopher and guide !!!!

***

ACKNOWLEDGMENTS

I am immensely grateful to the Almighty God for His kind "Grace". He has helped me conquer my fears with His guiding light, helping me achieve this Doctor of Philosophy Degree.

At this moment of accomplishment, I would like to express my deepest gratitude and appreciation to my supervisor Dr. Priyadarsi Nanda. This dissertation would not be possible without his able guidance and unwavering support. I am grateful for your expert feedback and valuable suggestions that helped me progress in my research. I am indebted to your kindness in offering your time and availability whenever I came to you for research discussion. Thank you for always being the extremely patient, positive, understanding, flexible and motivational guide, through out my research tenure. You gave me hope whenever I felt hopeless. Thank you for carving the researcher in me and showing me how to focus on seeking solutions to problems and always keeping a positive approach, while working on my research. I am blessed to have you as my mentor and my "Guru" for you have never given up on me even in times when I did give up on myself. I also wish to thank my co-supervisor Dr. Xiangjian He, for his valuable feedback and guidance during my research. Thank you for always offering kind and positive words of encouragement to keep me motivated in pursuing my research.

I would also like to thank my senior researchers Dr. Zhiyuan Tan, Dr. Aruna Jamdagni and Dr. Mohammed Ambu Saidi for their time and valuable guidance during the beginning of my research that helped me move ahead with my work. Thank you for all the knowledge sharing sessions that helped me build my research foundations. Many thanks to Eriyani and Aprillia from the School of Electrical and Data Engineering at the University of Technology Sydney for helping me with guidance with administrative tasks. I would also like to acknowledge Dr. Chandranath Adak for providing the PhD thesis template without which writing this thesis would have been an uphill cumbersome task. I would like to thank John Hazelton for his meticulous and prompt proof reading for my dissertation.

I am grateful for my time spent at UTS doing my research work as this gave me an opportunity to foster some great friendships. Big heartfelt thanks to Ashish Nanda and Amber Umair for always being there to inspire and motivate during my research. You both have helped me see the silver lining behind the dark clouds when I needed

it most. Thank you to Annie Baskaran, Nazar Waheed, Chau Nguyen, Nisha Malik, Madhumita Takalkar and Sara Farahmandian for their wonderful camaraderie, friendly banter, advice and support during my research and even outside. A very special thank you to my lovely friends Rattandeep Kaur, Sai Kiran Tadepalli (Sai Garu) and Ramya Bati for their nurturing friendships.

I take this opportunity to thank my father Dr. T.N.Nagar, and my mother Mrs. Shobha Nagar, for their unconditional love, trust, support, encouragement, guidance and motivation in my life, especially as I encountered challenges during my research journey. You have taught me the power of prayer and built in me the resilient, "never give up" attitude towards life. You inculcated values of dedication, perseverance, sincerity, integrity, truthfulness and loyalty. These golden values have made me accomplish my goals and become who I am today. You have always reinforced the belief in me that I can move mountains if I put my mind to it! Thank you from the bottom of my heart for being my safety net and giving me the mental strength and confidence to realize my dreams into reality. I owe you my existence and I can not thank God enough for blessing me with the most awesome and caring parents. This PhD has been made possible only because of your presence and relentless support in the final year of my research. I would like to dedicate my thesis to you both.

Special thanks to my younger sister, Kankana Nagar, for being the playful, happy person and also for being the wiser one between us. You always brighten up my moods and lift my spirits with your presence.

I thank my mother-in-law, Hurshit Nagar, for her love, best wishes and blessings. I would also like to fondly remember my late father-in-law, Hursh Nagar, whose blessings are always with me. He would be so proud of my big achievement.

Last but certainly not least, I would also like to thank my incredible husband, Prasannavadanam Nagar for believing in my dream and showering unconditional love and support during my research journey and in life. My heartfelt gratitude goes to my precious children and my forever cheerleaders, Aum and Anika. Their magical smiles always inspire me to keep giving my best in life.

Words fail to express my gratitude towards my extended family and all my family friends, who have always encouraged me to chase my research goals. I wish to conclude this acknowledgement, with a quote from one of my favourite books, "The Alchemist", by Paulo Coelho :

"And, when you want something, all the universe conspires in helping you to achieve it."

List of Publications

1. Nagar, U, Nanda, P & He, X Feature Analysis and Ensemble-based Intrusion Detection Scheme using CICIDS - 2017 dataset. (Submitted to Wiley, Concurrency and Computation Practice and Experience Journal.)

2. Nanda, P, Arain, A & Nagar, U 2019, 'Network Packet Breach Detection using Cognitive Techniques', Smart Systems and IoT: Innovations in Computing, Second International Conference on Smart IoT Systems - Innovations in Computing (SSIC-2019), Springer, Jaipur, India.

3. Nagar, U, Nanda, P, He, X & Tan, Z 2017, 'A Framework for Data Security in Cloud using Collaborative Intrusion Detection Scheme', SIN'17: Proceedings of the 10th International Conference on Security of Information and Networks, International Conference On Security Of Information And Networks, ACM Digital Library, Jaipur, India, pp. 188-193.

4. Ambusaidi, MA, He, X, Tan, Z, Nanda, P, Lu, L & Nagar, U 2014, 'A novel feature selection approach for intrusion detection data classification', 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE Computer Society, Beijing, pp. 82-89.

5. Tan, Z, Nagar, U, He, X, Nanda, P & Liu, R 2014, 'Enhancing Big Data Security with Collaborative Intrusion Detection', IEEE Cloud Computing Magazine, pp. 34-40.

TABLE OF CONTENTS

List of Figures

Figure **Page**

List of Tables

CHAPTER 1

INTRODUCTION

In our first chapter, we present an introduction and overview of the thesis. The research background along with the research motivation is presented here. This chapter also presents a summary of the contribution made through our research. In the beginning of the chapter, section 1.1 highlights the cyber security threats and concerns especially when majority of the software systems are functioning in cloud environment. We also mention in brief the relevance and issues of available benchmark datasets in testing the IDS solutions for prevalent and future threats and attacks. This chapter also provides some background information on the importance and need for intrusion detection systems in the field of cyber security. In section 1.2 we enlist the research motivation that propelled us to propose, develop, implement and test a new intrusion detection framework based on ensemble method using the latest available IDS dataset. We also state the problem statement. Section 1.3 provides the objectives of this dissertation and gives a summary of the research objects for this dissertation. Section 1.4 provides an overview of the contributions made by our research work. In the last section of this chapter, we give the structure of the chapters in the thesis.

1.1 Background

In today's day and age, internet has become an integral part of our modern society. Internet is an inevitable part of our world and human dependency on technology driven services and solutions has grown in leaps and bounds. According to statistics published

in [45] there are 4.13 billion Internet users across the world which is equivalent to more than half the number of the total world population using digital platforms to stay connected and contribute to the world economy. The forecast report from CISCO [44] states that the global population of internet users in 2021 is 58% which is an incremental rise as compared to 44% in 2016. With the rise and advance of digital technology, nations across the world have become technology savvy. This has lead to extensive use of internet for e-commerce, file sharing, social media, electronic communication via smart phones and smart devices. Individuals, government and private industry sectors, banking institutions, education, health, travel, tourism, hospitality and several prominent industries are now a part of the online, digital ecosystem.

Cloud computing represents one of the most significant shifts in information technology and is of great interest for academic researchers and the IT business community [122]. The advent of cloud technology has made the digital communities expand extensively since now there is no geographical limitation for businesses in expanding their ventures. The National Institute of Standards and Technology (NIST) provides the definition of cloud computing as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [110] [122]. The cloud computing delivery model offers computing as a utility, on demand and pay as you go service which is attractive for small and medium IT organizations to get on board. Cloud computing can drastically reduce the infrastructure overheads, reduce capital cost and focus on business competencies for organizations [122]. Statistics [162] indicate that in 2013 there were 2.4 billion users worldwide accessing cloud computing services and the number increased to approximately 3.6 billion in 2018. Authors [67] also noted that The US had the biggest market share for public cloud and its projected spending in 2019 was $124.6 billion. A press release by 451 Research [1] states that several business enterprises were able to sustain during COVID-19 pandemic due to their widespread adoption of cloud computing technology .

Another technology at the forefront of digital space is Internet-of-Things. As explained in [181] "IoT is essentially a platform where embedded devices are connected to the internet, so they can collect and exchange data with each other. It enables devices to interact, collaborate and, learn from each other's experiences just like humans do."

IoT devices are gaining popularity in the commercial consumer market as the share of wearable devices and smart home devices keeps increasing. IoT technology has contributed to the healthcare industry by offering wearable health monitoring devices and smart pacemakers. IoT contributes to the Vehicle to vehicle (V2V) communication in the transport industry and the infrastructure industry has advanced with smart cities. Digital control systems, statistical evaluation, smart agriculture, and industrial big data are offered by Industrial Internet of Things [104]. IoT technology is in the forefront for the military sector offering robots for surveillance and human-wearable bio metrics for combat [104]. Statistics [103] indicate that in 2019, there were nearly 26.6 billion active IoT devices and by 2025, it is predicted that approximately 75 billion IoT devices will be connected to the internet. Further to this, the same report forecasts that by 2026 the IoT device market is estimated to reach $1.1 trillion.

1.1.1 Overview of Cyber threats and attacks

Internet and electronic communication are the backbone of our society. However, there is a downside to this technological advancement. With the exponential increase of digital connectivity, cyber crime has been on the rise too. Criminal minds are continuously working in the background, exploring novel methods to attack the internet users, and hence, every device that is connected to the internet is under potential threat of being hacked or attacked. Malware, phishing, man-in-the-middle, ransomware, denial-of-service, SQL injection, DNS tunnelling, cryptojacking, zero day attacks are the most prevalent categories of cyber security attacks. The most common cyber attacks can be one or a combination of several of these categories of security attacks. The 2020 Data Breach Investigations Report [185] states that out of a total of 157,525 security incidents that were analysed, 3,950 were confirmed data breaches. An incident is defined as "A security event that compromises the integrity, confidentiality or availability of an information asset." in the report and a data breach is " An incident that results in the confirmed disclosure; not just potential exposure of data to an unauthorized party." Out of these 3,950 breaches, 45% featured hacking while 22% involved social attacks and 22% malware. Further, 37% involved user credential theft, 27% malware incidents were ransomware and 22% breaches were phishing based. 70% of these breaches had external perpetrators involved, 55% were by organized criminal groups and 30% were internal attackers [185]. Another security analysis [44] predicts data breaches to keep increasing and the total number of DDoS attacks are predicted to double from 7.9 million in 2018 to 15.4 million by 2023.

In the last few years, the online shopping trend has increased with e-commerce websites like amazon and ebay, bringing the world of consumerism to the personal desktop and mobile devices. Consumers can now shop for household items, devices and kitchen appliances, electronic gadgets, clothing, toys, entertainment, book travel tickets, holiday accommodation and so much more, right from the comfort of their living room. With online shopping becoming immensely popular, the vulnerabilities, threats and security risks of this new consumer trend have also been exposed. Symantec's Internet Threat report for 2019 highlights the rise in formjacking incidents where the customer credit card details and other important customer information are stolen using malicious JavaScript code on the billing page of the e-commerce websites. The stolen details are further sold in the cyber criminal market for huge monetary gains [168].

Cloud attacks have been a consistent threat to organizations and the internet community [46]. Gartner′s report mentions the evolution of cybersecurity threats and attacks and how before-mentioned attacks are still a relevant security challenge. The report further makes note of the importance of cyber security with the ongoing COVID-19 pandemic since the business world has changed dramatically and rapidly embraced digitalization with organizations moving to cloud computing [137]. The 2016 Internet security threat report by Symantec indicates that cloud computing systems are vulnerable to unauthorized access due to incorrect user configuration and bad data management [169]. A denial of service attack is conducted from a single internet connection which is used for inundating a victim's system with service requests and hence, exhausts the victim's system resources, which in turn makes the system unavailable to legitimate system requests and operations. Distributed denial of service attacks are an extension of DoS attacks and are more complicated to detect since the malicious requests on the victim's machine are simultaneously conducted via multiple internet connections and devices [47]. The famous DDoS attacks up till now include the Google attack in 2017 which was considered to be four times bigger than the one using Mirai botnet in 2016. The 2020 DDoS attack on Amazon Web Services is the latest and is considered the biggest to date [128].

The rise in malware related incidents indicates that computer systems still face the threat of malware attacks via email spamming, phishing. Incidents related to cryptocurrency mining malware and hacking are on the rise [185]. The data breach investigations

report 2020 published by Verizon [185] indicates that DoS is still the most prominent threat action variety followed by Social media Phishing and ransomware. In 2017, WannaCry malware caused much damage by exploiting a vulnerability in Microsoft Windows using the EternalBlue exploit. This malware encrypted the contents of the hard drives of compromised computers which were primarily used by United Kingdom's National Health Service [66]. If we consider the Australian cybersecurity landscape, the malicious actors are trying to exploit the COVID-19 situation to their maximum benefit and there has been a significant jump in the number of malicious cyber criminal activities [134]. Malicious actors are now more keen to attack corporate networks instead of individual servers, in order to gain illicit access to corporate data and sell the information on the dark web [161]. There has been a surge in coronvirus related malicious cyber activities. Symantec reported [167] approximately 5,000 malicious emails in the beginning of 2020 which had "coronavirus", "corona", or "COVID-19" as the subject line of the email and, as the pandemic cases increased, the number of such spam emails drastically climbed with approximately 82,000 emails in March 2020. Cyber criminals and threat parties have been using various methods to gain advantage of society's fear and uncertainty that has been created by the current global pandemic. These actors have employed email campaigns that use phishing emails, malspam emails, and scams to trick users into buying COVID-19 related products like masks, medical equipment and immunity boosting substances. Compromised domains are used to build phishing URLs that lead to stealing of user credentials. Phishing emails related to COVID-19 related fund raising are being circulated. These emails urge the user to review a document that provides the details of the funding and prompts for user credentials to access the document. Emails that appear to be from legitimate organisations like World Health Organization (WHO) have been circulating which lure the user to reply to these emails in order to access financial funds for preventing disease. Eventually the user is tricked to pay a fee for releasing this fund. Further more, fake emails from WHO have been sent indicating that there has been breakthrough research in combating COVID-19 and prompts the user to click on a file or PDF which leads to downloading malicious content on the user machine. Extortion emails related to COVID-19 have also seen a rise, where the sender, claiming to be a neighbor, indicates that they have acquired COVID-19 virus and will not survive. The recipient is threatened to be infected unless money is deposited to the Bitcoin wallet address in the email [178].

1.1.2 Intrusion Detection System - Need for robust cyber security

A variety of schemes and tools are available to detect cyber attacks and block them. Firewalls have been used as one of the tools for cyber security. A firewall is placed between the network to be protected or a network that has been labelled as the trusted zone and the outside network like Internet. The function of a firewall is to check every packet that is incoming or outgoing for the trusted network zone and, based on the predefined firewall rules, if there is a match or mismatch, take an action to accept or discard the particular packet [72]. It is usually used as the first line of defence; however, the firewall does not provide enough security from zero day attacks and sophisticated cyber security attacks. Majority of the organisations employ intrusion detection systems as a second wall of security after deploying a firewall. There is a very large amount of research done in the area of intrusion detection and even now researchers continue to work on various IDS schemes.

Cloud computing represents one of the most significant shifts in information technology and is of great interest for academic researchers and the IT business community. The National Institute of Standards and Technology (NIST) [110] provides the definition of cloud computing as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [122]. The cloud computing delivery model offers computing as an utility service, which can be provided on demand. The pay as you go billing service model offered by the cloud providers, is very attractive for small and medium IT organizations to get on board. Cloud computing can drastically reduce the infrastructure overheads, reduce capital cost and focus on business competencies for organizations. As much as the popularity of cloud is increasing, it is observed that potential customers are still sitting on the fence. They seem to be apprehensive and nervous in adopting cloud computing for their organization mainly due to the security gaps in cloud. Hence, the importance of cloud computing security cannot be ignored. Cloud computing security deals with securing various aspects of security comprised of data transparency, data authenticity, data authorization etc..It essentially covers the existing security challenges that are extended to the cloud as well as security risks that are introduced due to the unique nature of cloud architecture and deployment methods

[122].

Several researchers [17][75][84][89][102][107][114][144][166] [186] have discussed the security threats and risks within the cloud and proposed approaches to make cloud computing a secure and trustworthy domain. However, there is much more to be achieved, in order to ensure that the fears of potential cloud customers are addressed particularly in the face of new applications and the ways data are accessed [122].

The threats and risks in cloud computing are discussed further in the following sections. The main cause of concern in the cloud is the lack of user control over their data once it propagates to the cloud. Users are unsure of the level of accessibility of the cloud provider to their data. For an organization / cloud customer, this is the biggest obstacle in the cloud domain as the cloud customer does not want the cloud provider to access their personal and sensitive data and manipulate it for monetary gains or malicious acts. The security measures currently in place do not leave much control in the hands of the users and thus makes them feel vulnerable and unsafe in the cloud. The data owners have very little choice but to trust the cloud providers and the security they offer in the cloud. Further, data owners have little or no visibility of the security and processes once their data is uploaded to the cloud and hence, feel dependent upon the cloud provider [122].

A private cloud has multiple entry points and multiple active instances at a given point of time. Hence, any potential intruder has several pathways to launch an attack in the cloud. These intrusions can be coordinated and spread out in parts over the various VMs in the cloud; hence, they can go undetected by the conventional standalone HIDS or NIDS. The threat actors are further getting sophisticated with their attack mechanism using new advance technologies. Attackers make use of distributed attack mechanisms using botnet to launch simultaneous botnet attacks at multiple network entry points. From an individual IDS view, the network activity seems normal and hence, the intrusion is not detected. However, the aim of these distributed attacks is the eventually target a single victim machine / network after crossing the initial entry point. This is a gap for enterprise and collaborative networks like cloud as these co operative intrusion attacks can target and shut down the services offered by cloud [173].

Intrusion detection can be defined as “the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, which attempts to protect confidentiality, integrity, availability, or even bypassing the security mechanisms of a computer or network"[15]. A generalized framework for a conventional cybersecurity system is depicted in Figure 1.1 and consists of network and host defense systems from cybersecurity threats, which we discussed earlier in this chapter.

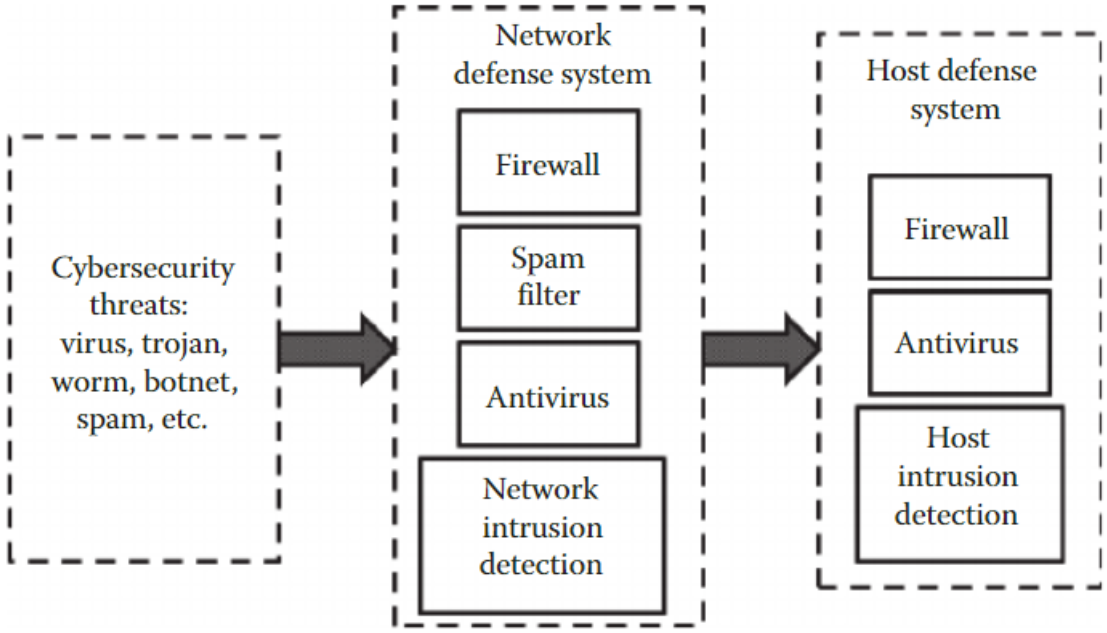

Figure 1.1: A generalized framework for a conventional cybersecurity system [57]

IDS can be based on software or hardware that is strategically placed at appropriate detection points in a system or network and automatically detects possible intrusion attempts and protects them from any future attacks.

IDS can be classified as Host-based Intrusion Detection Systems (HIDS), Network-based Intrusion Detection System (NIDS) and Distributed Intrusion Detection System (DIDS). This classification is based on the location point of the IDS. A Host based Intrusion Detection System (HIDS) is deployed on an individual host or machine that needs to be monitored for intrusion. HIDS continuously monitors the host system activities such as the file system being used, the network events occurring on the host, the system calls being used. It also monitors the host system for any changes made to the host kernel, host file system or the program behaviour. The HIDS reports an intrusion, if there is any

deviation from the normal expected system behaviour of the host device. HIDS efficiency is dependent on the characteristics that are defined to be monitored for the given host system[112].

A Network based Intrusion Detection System is used to monitor and collect network traffic with the aim of spotting malicious events that may be attacks like DDoS, DoS, ARP spoofing, port scans. This detection is done by comparing the current network behaviour and the previously observed normal network behaviour[112]. NIDS are most commonly deployed at the network gateway and routers, since these are the main points for entry/exit of network traffic in any given network[111].

Distributed Intrusion Detection System (DIDS) is also known as Collaborative Intrusion Detection System (CIDS). In a CIDS, there are multiple IDS; either HIDS or NIDS, each of which detects the intrusive activity and further communicates the same to a central control system to record an intrusion alert. The central control system is responsible for the analyses of the reported alerts and also correlates the alerts collected from different IDSs in the CIDS topology. Based on the analysis of the central control system, further action can be communicated to other IDS in the network, which leads to a comprehensive intrusion detection system.

Based on the detection methodology, IDS can be classified primarily as Signature based IDS and Anomaly based IDS. Signature based IDS are also known as Misuse detection. Signature based intrusion detection method uses a signature database which has a defined set of rules / signatures [112] for comparing the current signature with a previously defined intrusion signature. An alarm is triggered if there is a match of current signature with the signature database. SIDS has great detection accuracy and low false positives. Further, the IDS based on this detection method is easy to maintain [112]. However the main drawback of this methodology is that the signature database needs to be updated regularly in order to detect new intrusion attacks. SIDS fail to detect zero day novel attacks as there is no signature in the database for the same. SNORT and Bro are the most commonly used open source tools that are based on Signature based intrusion detection technique [90]. A number of researchers have done some useful work on using signature based IDS in the cloud [17][102][107][145]. It is noted that signature based IDS can be positioned at the front end in the cloud (at VMs), enabling the detection of outsider attacks. It can also be positioned at the back end of the cloud for detecting

internal or external intrusion [122].

Anomaly based Intrusion Detection Systems focus on detecting any anomaly in the system behaviour with respect to defined normal system behaviour. Anomaly based IDS schemes involve a training and testing phase. During the training phase, the model is trained to learn the normal system behaviour and later during the testing phase, this model is used to test a new system dataset. Based on the trained model, the IDS predicts intrusion activity for the events that show anomalous behaviour compared to the trained normal user profile[90]. The main advantage of anomaly based detection technique is its ability to detect zero day attacks. However anomaly based IDS suffer from high false positives since a normal user behaviour which is new to the trained model can be classified as an intrusion.

IDS can be further classified based on the deployment methods such as; software based IDS, hardware based IDS or VM based IDS[144]. A number of authors have proposed several intrusion detection systems that are based on anomaly detection technique[53][68][75][186]. Anomaly detection can be applied in the cloud environment at various layers of the cloud architecture, which makes it a challenge to monitor intrusions over multiple layers of the cloud[113] [122].

1.1.3 Research Motivation : Challenges with Intrusion Detection schemes, Feature selection techniques and IDS benchmark datasets

Intrusion detection systems are still a very popular research area amongst academics and the research community and several approaches have been explored with the main aim of increasing the detection efficiency of the IDS. Machine learning methods have been very popular in implementing IDS schemes; however, these techniques suffer from the limitation of the single machine learning classifier being weak which impacts the IDS detection rate [98]. Further, most of the research schemes mentioned in the literature have not considered the time taken for implementing the machine learning scheme. This is a vital feature since a highly efficient scheme with high detection rate may suffer from a slow rate of model generation [193].

In addition to this, the IDS datasets available for machine learning model evaluations suffer from high dimensionality which can be detrimental for the computational and resource overheads for fast and efficient intrusion detection. Feature selection methods have been employed by researchers working on machine learning IDS schemes [38] [86] [96] [182]. Ensemble-based machine learning methods combine multiple base learning methods to derive an optimum model for prediction and are used to overcome the shortcomings of weak classifiers. There is much research work done using ensemble-based feature selection methods for data mining applications such as medical sciences, pattern analysis and machine intelligence areas [142]. However, ensemble-based feature selection technique is relatively new within the intrusion detection domain and, as is highlighted by [26] there are only a handful of feature selection approaches, including [26] and [193], that have considered using ensemble-based feature selection methods for dimensionality reduction in the intrusion detection area. Hence, there is a need to further investigate feature selection using ensemble techniques for the sphere of intrusion detection.

Another pressing issue with IDSs is the lack of benchmark datasets that reflect the latest intrusion threats and attack scenarios for testing and validation of IDS schemes. The available benchmark datasets used for training and testing of intrusion detection systems do not cater to the enhanced attack scenarios and zero day attacks. Hence, there is a gap and a major need for a new benchmark dataset that caters to all the requirements for benchmark IDS dataset for training and testing. The publicly available data sets with which researchers can experiment the intrusion detection schemes include DARPA, KDD'99, DEFCON, CAIDA, LNML, CDX, Kyoto, Twente, UMASS, ISCX2012 and ADFA. These datasets have often been criticized by the research community since they do not cover diverse traffic and comprehensive attack profile [21]. Hence, these datasets cannot be relied upon to test the intrusion detection schemes in modern world. Moreover, the nature and execution of intrusion attacks has evolved with victims being subjected to diverse attacks.

1.2 Problem Statement and Research Contribution

This thesis primarily addresses three main gaps that we have uncovered during our research and we aim to address these issues with our research work. We explain the

problem statement and also mention our research contributions aimed to address the identified issues as follows :

1. Feature selection techniques are very popular and exhaustively employed in IDS research for reducing the size of IDS datasets. However researchers have given considerable focus on single method feature selection while emphasis on ensemble based feature selection methods has been not been extensive. We address this gap by developing an ensemble based feature selection technique which uses Information Gain, Correlation and Particle Swarm Optimization (PSO) methods for the high dimensionality IDS datasets namely CICIDS - 2017 and NSL - KDD.

2. The second issue investigated and addressed by this thesis is the limitation of machine learning detection methods with a single weak classifier resulting in less than optimum detection rate and high false alarm rate. Further the issue of time taken to build the intrusion detection model is critical to an IDS scheme. We address this challenge in our dissertation by developing an ensemble based intrusion detection scheme which combines multiple classifiers (KNN, C4.5 and Random Forest) into a single ensemble classifier using the majority voting method which is based on the average of probabilities (AOP) combination rule. This is the second contribution of our research work.

3. The publicly available bench-marked intrusion detection datasets such as DARPA, KDD 99 and NSL - KDD, are highly outdated with respect to the threat scenarios included and the network systems they represent [108]. However due to the lack of any comprehensive IDS datasets which represent modern attacks and network settings, these old datasets are still being used in validation of IDS schemes by the research community. Recently, the University of New Burnswick has publicly published their new CICIDS - 2017 dataset. CICIDS - 2017 has been created by capturing complete traffic using 12 different machines set up within the Victim Network and attacks are launched from the Attack Network. CICIDS - 2017 dataset includes all available protocols like http, https, SSH, FTP and SMTP. This dataset has been generated taking into consideration the latest and most relevant attack scenarios like DoS, DDoS, Brute Force, Web Brute force, XSS, SQL injection, Heartbleed attack, slowloris, slowhttptest, Hulk DDoS attack, GoldenEye attack,

infiltration, Port scan and Botnet. This new dataset addresses important evaluation criteria such as complete network configuration, complete traffic, labelled dataset, complete interaction, complete capture, available protocols, attack diversity, heterogeneity, feature set and meta data [152]. However, we have found that not many research works have used CICIDS - 2017 dataset for IDS evaluation yet, due to the dataset being comparatively new and hence, does not have sufficient performance bench-marking using feature selection. As our third contribution, this research aims to benchmark CICIDS - 2017 dataset using our proposed ensemble-based feature selection technique and ensemble based intrusion detection framework. We further compare the performance of our proposed IDS model with the traditional NSL - KDD dataset.

1.3 Research Overview : Objectives and Methodology

The research objectives and research methodology employed in our thesis are explained as follows :

1.3.1 Research Objectives

The objective of this thesis is three-fold. The high level focus areas of this research work are listed as follows :

1. To develop a novel ensemble based feature analysis model which employs the feature selection methods based on information, correlation and swarm intelligence and combine the chosen features from each of these methods to further combine and generate a unique feature subset for the chosen IDS dataset.

2. To develop a novel ensemble based intrusion detection model to address the drawback of single machine learning classifier and hence, provide superior intrusion detection performance which is comparable to the state of art IDS schemes available in literature.

3. We further aim to benchmark the performance of latest publicly available intrusion detection dataset CICIDS 2017, using the proposed Ensemble based feature selection and intrusion detection framework mentioned in 1. and 2.

We explain below, the research methodology followed during our research work to achieve the research objectives listed in section 1.3.1 above.

1.3.2 Research Methodology

The research methodology used to address the research challenges can be summarized as below :

1. To conduct an extensive literature review for intrusion detection systems and the publicly available IDS datasets. This review would also include techniques used for feature selection and anomaly detection. The aim of this literature review is to uncover gaps in feature selection and intrusion detection research which can be further used for developing our proposed solutions. (This is discussed in Chapter 2.)

2. To investigate various Feature selection methods and propose a novel ensemble-based feature selection approach to generate a set of optimum features for CICIDS - 2017 datasets. (This is discussed in Chapter 3.)

3. To investigate the historical benchmark IDS datasets namely NSL - KDD and UNSW - NB15 and further develop a comprehensive understanding of the latest CICIDS - 2017 dataset features. (This is discussed in Chapter 3.)

4. To study the state of the art machine learning methods employed for intrusion detection and further explain the importance of ensemble-based intrusion detection approach. (This is further discussed in Chapter 4.)

5. To design a novel Ensemble based intrusion detection scheme using three classifiers namely K-Nearest Neighbor (KNN), C4.5/J48 and Random Forest (RF). (This is discussed in Chapter 4.)

6. To implement the proposed ensemble-based feature selection scheme for IDS datasets namely CICIDS - 2017, UNSW - NB15 and NSL - KDD and generate optimal features which will help in reducing the size of the aforementioned datasets. (This is discussed in Chapter 5.)

7. To implement the proposed ensemble-based intrusion detection scheme for IDS datasets namely CICIDS - 2017, UNSW - NB15 and NSL - KDD using original dataset features and the ensemble-based features generated using our proposed feature selection scheme. (This is discussed in Chapter 5.)

8. To benchmark the performance of CICIDS - 2017 dataset using the performance metrics calculated based on the results of our implementations. We also need to observe the time taken for generation of our proposed ensemble based IDS model. (This is discussed in Chapter 5.)

9. To summarize and conclude our thesis. (This is discussed in Chapter 6.)

1.4 Thesis Organization

This thesis has been organized as below:

The first chapter provides the introduction and background of the cyber security threat and attack landscape globally. On the basis of this overview we establish the importance of an efficient intrusion detection system for detecting a variety of cyber security intrusion attacks. We also highlight the challenges involved in intrusion detection methods and the lack of availability of benchmark IDS datasets which include the modern day attack scenarios for IDS validation. Based on the challenges discussed, we formulate our problem statement and also list our main contributions. We finally provide an overview of our research objectives in this chapter.

In Chapter 2, we present an in-depth literature review of state of the art in intrusion detection systems in general and we also review the existing CIDS schemes in cloud computing environment. We also present a collaborative intrusion detection framework for cloud computing. We further provide the design of a CIDS system model using open sourced HIDS and NIDS tools for intrusion detection and SIEM tools for alert logging and analysis. We further review the various machine learning algorithms widely used for intrusion detection and focus extensively on ensemble based techniques specifically. We also explain how our work contributes further to this area of research. In addition, we review the existing historical benchmark intrusion detection datasets that are publicly available and their use in validation of IDS models. We also conduct a literature review of the latest publicly available CICIDS - 2017 dataset and highlight our contribution in using the same for validation of our proposed model. Finally, we conduct a literature review for feature selection methods employed in intrusion detection systems.

In Chapter 3, we discuss the CICIDS -2017, UNSW - NB15 and NSL - KDD datasets in detail and explain the data pre-processing activities employed to prepare datasets

for our experiments. We propose our ensemble-based feature selection framework and explain the feature selection process. We also provide a brief theoretical background for the feature selection algorithms used in our framework. In addition, we explain the evaluation metrics used in our research for validation of our proposed IDS scheme.

In Chapter 4, we focus on the increasing popularity of ensemble based methods in intrusion detection models. We propose our novel IDS model based on Ensemble based technique using multiple classifiers (KNN, C4.5/J48 and Random Forest). We provide the experimental design of our proposed ensemble based IDS scheme and explain the various components our proposed ensemble-based IDS scheme.

In Chapter 5, we implement the ensemble-based feature selection method (proposed in Chapter 3) for the CICIDS - 2017, UNSW - NB15 and NSL - KDD and generate optimum feature subset for each of these datasets. We further implement our proposed ensemble-based intrusion detection framework (proposed in Chapter 4) using the original features for each of the aforementioned datasets. We again run the experiments for all three datasets, using the features generated using our feature selection technique. The results of these experiments are used to calculate the performance metrics and results are reported. Furthermore, comparative analysis is provided and we discuss the performance our our scheme and explain how the proposed ensemble-based framework provides the performance benchmark for CICIDS - 2017 dataset.

Finally, in Chapter 6 we provide a conclusion of our thesis. We include the key findings and also suggest pointers for future work.

CHAPTER 2

INTRUSION DETECTION SYSTEMS - TAXONOMY AND RELATED WORKS

Intrusion detection is a much researched area in cyber security domain. As discussed in the previous chapter, cyber intrusions and hacking activities never seem to subside and are on the rise with technology advancement in the cyber world. The relevance of intrusion detection systems in the current security paradigm cannot be understated. In this chapter we dig deeper into the concepts of intrusion detection systems and further discuss the commonly prevailing threats. We also briefly look at the limitations and challenges faced by current readily available research IDS datasets. The chapter concludes with a discussion of the literature review in the area of intrusion detection.

2.1 Taxonomy of Intrusion Detection Systems

Research in the field of intrusion detection systems dates far back in the history of cyber security. Several research survey and review papers have been published by the academic researchers and industry in this area. We now discuss the basic taxonomy of the intrusion detection system. Any activity that leads to the loss, damage or incorrect functioning of any information system, is considered as an act of intrusion [90]. Intrusion detection system or IDS is a critical security component which primarily aims to discover,

determine and identify any misuse of data, any data duplication or alteration and any damage or destruction of the data or software information system [184].

Denning et.al. [55], have enlisted the requirements for an intrusion detection system which are briefly explained as follows. Firstly, IDS should be capable of detecting all possible intrusions and potential network threats that may occur. We discuss the prominent threats in section 2.2 of this chapter. Applicability is listed as a second requirement for IDS and requires the IDS to be adaptable to any hardware, operating system or the environment. IDS design and software should be independent of the application. High rate of detection and low false alarm rates are described as another essential feature required to be incorporated while designing an IDS. Further, ease of use and modifiability are listed as desired characteristics for an intrusion detection system. The IDS should be user - friendly and at the same time should be easy to maintain, update and modify by security professional and administrators. Also, a good IDS should be self-learning and the IDS database should enforce integrity and security from spoofing or DoS attacks. Some other features desirable for an intrusion detection system also include prediction performance, time performance and fault tolerance [95].

Deber et.al [54] and Stefan Axelsson [14] were a few of the first researchers who provided the classification of an intrusion detection systems in a comprehensive manner. Figure 2.1 depicts the elemental taxonomy of an intrusion detection system. These categories are further explained in following sections of this chapter.

Primarily intrusion detection systems can be classified into three types based on the data source as Host-based IDS, Network-based IDS and Hybrid IDS. They can also be classified based on what the detection method is employed. These types are Signature-based IDS and Anomaly- based IDS. They are further classified as Real time IDS and Offline IDS based on the data audit time. According to the system architecture, IDS can be either Distributed or Centralized types. Finally, IDS can also be classified based on the action taken after the intrusion has been detected. These can be either active IDS or passive IDS.

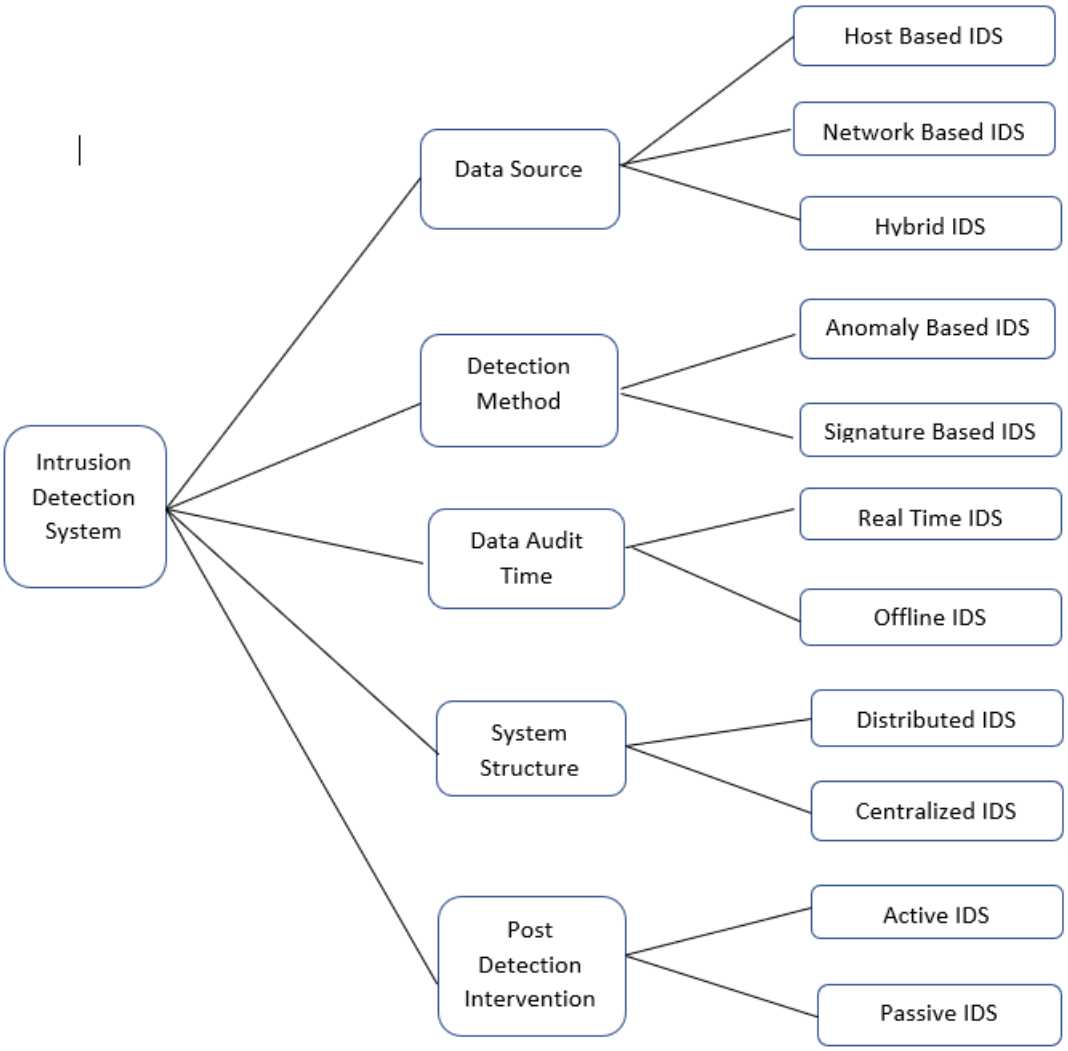

Figure 2.1: Taxonomy of Intrusion Detection Systems.

2.1.1 Intrusion Detection System Based on Data sources

As shown in 2.1, IDSs can be classified based on the data source i.e. the location at which they monitor the intrusion in an information system network. Based on this classification there are two types of IDSs : host-based IDS and network based IDS.

Host-Based IDS Host-based IDSs (HIDS) are deployed on individual or several host machines in a computer system. The aim of HIDS is to monitor the host device for any intrusive activity and report the same. HIDS monitor and inspect data from the host such as log files and system calls. HIDS are capable of detecting any insider attacks

being executed from the host machine.

Network-Based IDS On the other hand, network-based IDS (NIDS) are deployed at the network nodes of the information system and they monitor the incoming network traffic. For a large network topology, the NIDS can be deployed at several network nodes and hence, can monitor the complete incoming network traffic and detect any network initiated attacks from the outside world. The various advantages and disadvantages of HIDS and NIDS are listed in 2.2.

Hybrid IDS As shown in 2.2, when HIDS or NIDS are used as standalone detection engines, they do not provide a multi-tier intrusion detection. HIDS can only detect insider attacks from the host while NIDS can identify outsider attacks from the network. For an efficient, all round IDS, several researchers have proposed IDSs which combine the both HIDS and NIDS. Such a classification of IDS where both HIDS and NIDS are used together for intrusion detection, are called hybrid intrusion detection systems. Hybrid IDS have the added advantage of both HIDS and NIDS and hence, provide a complete intrusion detection system from both the insider and outsider attacks.

	Advantages	Disadvantages	Data source
HIDS	• HIDS can check end-to-end encrypted communications behaviour. • No extra hardware required. • Detects intrusions by checking hosts file system, system calls or network events. • Every packet is reassembled • Looks at the entire item, not streams only	• Delays in reporting attacks • Consumes host resources • Needs to be installed on each host. • It can monitor attacks only on the machine where it is installed.	• Audits records, log files, Application Program Interface (API), rule patterns, system calls.
NIDS	•Detects attacks by checking network packets. •Not required to install on each host. •Can check various hosts at the same period. •Capable of detecting the broadest ranges of network protocols	•Challenge is to identify attacks from encrypted traffic. •Dedicated hardware is required. •It supports only identification of network attacks. •Difficult to analysis high-speed network. •The most serious threat is the insider attack.	•Simple Network Management Protocol (SNMP) •Network packets (TCP/UDP/ICMP), •Management Information Base (MIB) •Router NetFlow records

Figure 2.2: Comparison of IDS technology types based on their positioning within the computer system [90]

2.1.2 Intrusion Detection System Based on Detection Methodologies

Based on the detection method, IDS systems can be classified into two types; Signature based Intrusion Detection System (SIDS) and Anomaly based Intrusion Detection System (AIDS).

Signature Based Intrusion Detection System Signature based intrusion detection systems (SIDS) are based on detecting the known attacks by comparing the attack signature with the signature database of the SIDS. SIDS has a database of previously known attack signatures. Using pattern recognition methods, signature of a known attack is compared with the signature database of the SIDS and if there is a match, an intrusion is detected and an alert is registered in the alert logging system of the IDS. Several research papers in literature mention SIDS as Knowledge based detection and also as Misuse Detection [90]. SIDS are efficient and effective and have an excellent detection accuracy for attacks that are known. However, SIDS cannot detect zero-days attacks or any attack whose signature is not in the SIDS rule database. Another drawback that SIDS suffer from is the frequent updating of the signature rule database to keep it up to date with the latest attacks. This process can be tedious and time consuming.

Anomaly Based Intrusion Detection System Anomaly based Intrusion Detection Systems (AIDS) have been an ongoing area of study with academic researchers. AIDS works on the concept of detecting behaviour in the incoming traffic that significantly deviates from the normal observed behaviour of the network. These deviations can also be called anomalies or aberrations [37]. An AIDS model development is made of a training phase and testing phase. In the training phase, normal data traffic profile is used to train the AIDS model with the aim to make it learn what is the normal behaviour. The testing data is a separate set of data observations that are used to test the above trained AIDS model to confirm the accuracy of intrusion detection [90].

AIDS resolve the issue of zero-day attack detection which is suffered by SIDS since they do not use any signature comparison methods for intrusion detection. However, AIDS have some drawbacks too. AIDS suffer from high false positive rates. This is because the new behaviour that deviates from the normal trained profile for AIDS, can be a genuine intrusion patter or it can be a new normal activity that has not been seen

before for the given network. Hence, AIDS cannot differentiate between the two and raises an intrusion alert, which leads to high false alarm rates.

2.1.3 Intrusion Detection System Based on System Structure

Another group of intrusion detection system classification is based on the organization of the IDS system structure. This type of classification has two types, centralized Intrusion detection system and Distributed intrusion detection system. These are briefly explained below.

Centralized Intrusion Detection System Centralized intrusion detection systems consists of multiple IDSs which can be either HIDS monitoring the traffic at the host or NIDS monitoring the network traffic. These IDS further share the alerts or the traffic details with a central unit that is responsible for further analysis. The major disadvantage of Centralized intrusion detection systems remains the possibility of a single point of failure as the important task of analysing the reported alerts or the extracted data from the network is performed here. Also the increase in the number of monitoring IDSs in an expanding information system network creates scalability issues [183].

Distributed Intrusion Detection System Distributed IDS usually perform the function as a monitoring device and also serve as the analysis unit, thus each IDS shares the jobs individually. Distributed IDS, hence, can make local decisions regarding the intrusions and can take actions based on the output of the analysis unit. However, since each of the IDS in a distributed architecture is not aware of the state of the remaining network, information circulation between the distributed IDS can be challenging. In order to plug this gap, distributed IDSs use Peer-to-Peer (P2P) set up for exchange of information amongst themselves [183].

Figure 2.3 explains the concept of centralized IDS and distributed IDS as explained in [183]. As shown in the figure, for centralized IDS architecture, M represents the IDSs that function as the monitors as well as intrusion detector units and A is the central analysis unit. In a distributed IDS architecture, the individual IDSs serve as monitoring, detecting and analysis units.

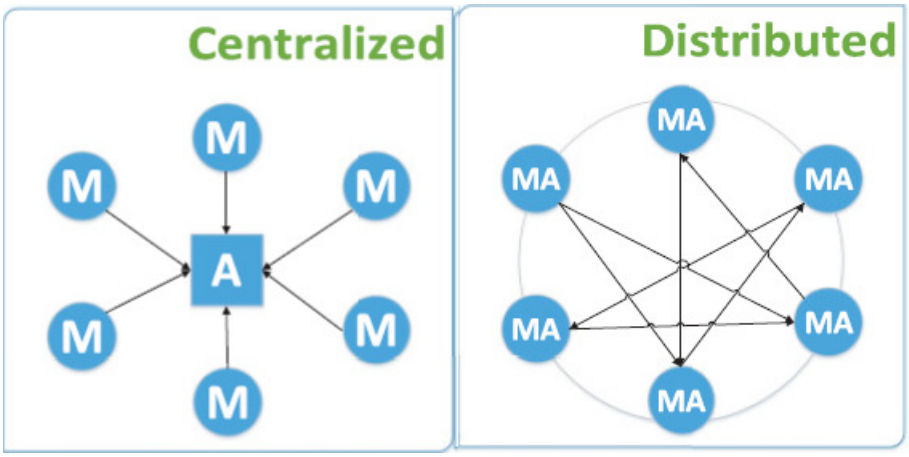

Figure 2.3: Centralized and distributed IDS (M=Monitors and A=Analysis units) [183]

2.1.4 Intrusion Detection System Based on Audit Time of Data

Intrusion detection systems can be classified into two types based on what time the data is audited. These can be either real time intrusion detection systems or offline intrusion detection systems.

Real-time Intrusion Detection System Real-time intrusion detection systems, as the name suggests, are capable of reporting the intrusions when they occur. Hence, when the attack is detected, the alert is raised at the same time. SNORT [157] is a freely available signature based network intrusion detection system that can generate alerts in real time and store the same in the log file. Another open source real time IDS is OSSEC [131] which is used as a host-based IDS and can be used as both signature based IDS and anomaly based IDS. In one of our initial studies during our research work we have proposed a hybrid intrusion detection system framework using SNORT and OSSEC tools, which will be explained in the following section of this chapter.

Off-line Intrusion Detection System Off-line intrusion detection system cannot raise alerts at the time of attack. Rather, in this case, as the name implies, the attack data is lodged in log files which are further used for analysis and data recovery at a predefined time later on.

2.1.5 Intrusion Detection System Based on Action taken after detection

Intrusion detection systems can also be classified on the basis of action that is taken by the IDS after detecting the attack. Again, there are two types of IDS under this group : Active intrusion detection system and Passive intrusion detection system,

Active Intrusion Detection System Active intrusion detection systems are capable of monitoring the network traffic and analyzing it for any intrusive pattern. If an intrusion is detected, the active IDS can take preventive actions to block and mitigate the attack from progressing further without any manual intervention. Active intrusion systems are also known as Intrusion Detection and Prevention systems (IDPS).

Passive Intrusion Detection System Passive intrusion detection systems, unlike active IDS, do not execute any real time action to block the malicious behaviour. Instead, passive IDS raises an alert whenever there is an attack detected, which warns the network administrator of the intrusion. Hence, passive intrusion detection systems require manual intervention to decide the action whenever an intrusion is detected and an alarm is raised.

2.1.6 IDS Based on Implementing Approaches

One way to classify detection approaches can be seen as based on anomaly detection of misuse detection. [98] list five subcategories based on these detection characteristics. These can be applied standalone as well as in hybrid form:

(1) Statistic-based approaches are based on statistical parameters such as mean and standard deviation. They also involve calculation of intrusion probability and use a predefined threshold value to determine intrusions.

(2) Pattern-based approaches use pattern recognition techniques where string matching is used to identify intrusions after comparing with known attacks.

(3) Rule-based methods use conditional statements such as IF-THEN or IF-THEN-ELSE to create rules in order to build the normal profile for detection model.

(4) State-based methods use network behaviours to build a finite state machine which is used to detect attacks.

(5) Heuristic-based approaches are built upon biological concepts and combine artificial intelligence to arrive on the detection engine.

The taxonomy of anomaly detection has been much written about in several research review and survey papers. However, anomaly detection techniques are quite widespread and there is no one best way to define the same. We found that the anomaly detection methods can be classified in more that one way and there is no single best approach to provide the anomaly detection taxonomy. As an example some published research survey or review papers [90] [4] [95] mention common anomaly detection techniques; however, the groups that have been used to classify these detection methods, may overlap or be entirely different across these papers. Khraisat et al. have listed three categories for AIDS methods which are (1) Statistics based (2) Knowledge based and (3) Machine learning based. They give examples of each group which include uni variate, multivariate and time series model for statistics based methods, finite state machine, description languages and expert systems for Knowledge based. They describe techniques like decision trees, Naive Bayes networks, genetic algorithms (GA), Hidden Markov model, fuzzy logic and KNN.

One of the survey papers [116] by Moustafa et. al. provides another comprehensive taxonomy for network anomaly detection as described in figure 2.4. According to their classification the network anomaly detection approaches or techniques can be grouped under six categories. The first is classification-based which includes Support Vector machine (SVM), Artificial Neural Network (ANN) and K- Nearest Neighbour (KNN). Statistical-based approaches can be further sub divided into Parametric and non-parametric types. Next are clustering-based methods which can involve regular

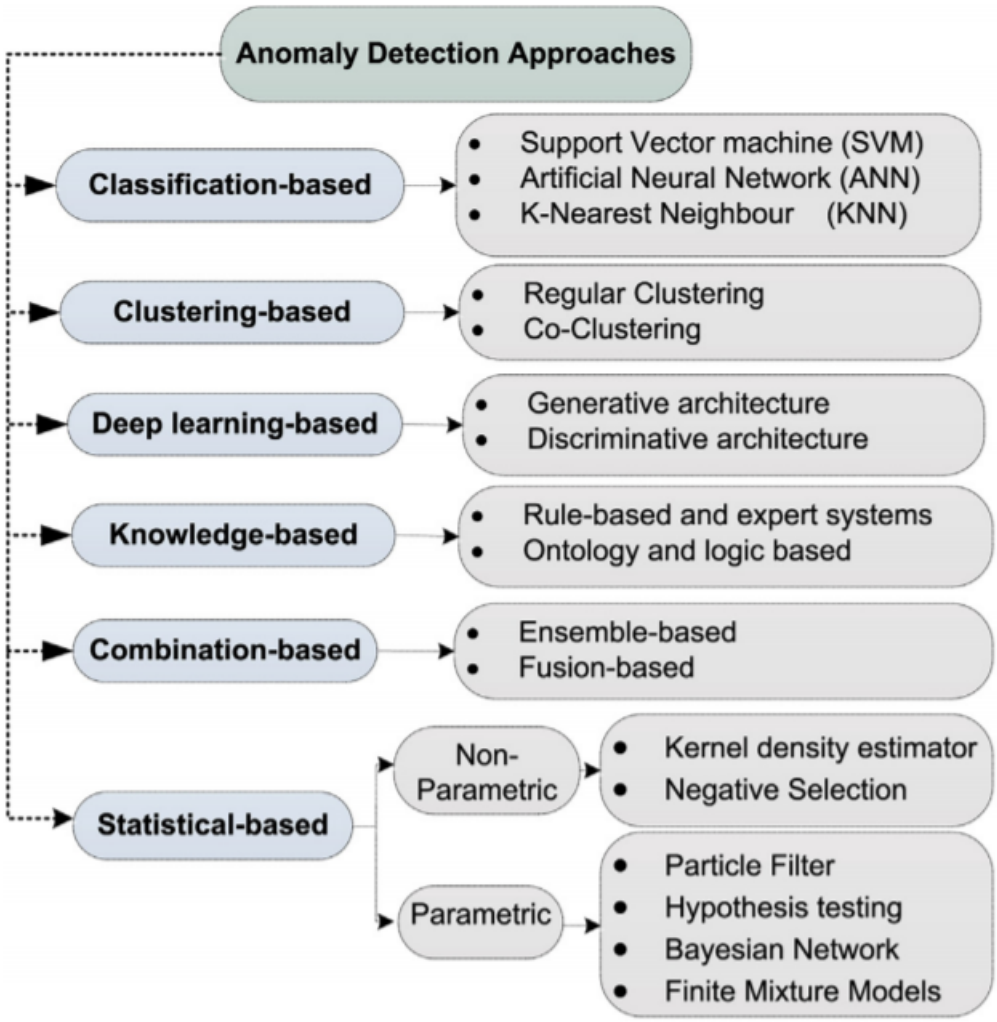

Figure 2.4: Taxonomy of network anomaly detection approaches [116]

clustering or co-clustering. Deep learning-based and Knowledge-based approaches are also widely used. Recently, combination-based approaches are being extensively used by researches trying to propose novel detection methods. These involve Ensemble bases and Fusion based approaches. Hence, the understanding of the taxonomy of anomaly detection in network intrusion is fundamental to our research. One of the main contributions of our research in this thesis is to propose a novel ensemble-based approach for network intrusion detection and to build the IDS based on our proposed scheme.

2.2 Taxonomy of Network Threats

Cyber attacks can be classified in several ways. Anderson [11] has identified the attacks as internal penetration, also known as insider attacks and external penetration which are also called outsider attacks. Insider attacks are executed by a perpetrator who may have access to the information systems but does not have administrative privileges or is not an authorized used of the computer systems. On the other hand, an outsider attack is when the attackers do not have physical access to the computer systems, but gain access via wireless sources such as the internet. Some researchers [79] have identified the attacks based on the activities and the attack targets. They have classified the network threats on the basis of the threat source, what OSI layer is being affected and if the attacks are active or passive. The threat sources identified and discussed further include network threats, host threats, software threats, physical threats and human threats.

Intrusions in information systems have further been broadly classified into numerous classes [25] [90] [95] [116] as follows:

Virus - Virus is a software program that infects the target system by duplicating itself without the knowledge of the target user. Virus can easily target and infect one system in the network, which if is connected to other machines in the network, can progressively pass on the infection to multiple targets without being detected.

Worm - Worm is also a self replicating software program that can reproduce onto the host or the network services in a computer information system. This attack, though not as potent, is capable of serious damage to the target network as it uses high network bandwidth.

Trojan - Trojan programs cannot be detected in plain sight as they appear to be useful applications. However, they may be carrying disruptive codes that potentially aim to create backdoor entry into the system and let the attacker gain control of the system without any user authentication.

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) - The aim of Denial-of-Service (DoS) attacks is to block legitimate users from accessing systems and resources like web services or data. This is achieved by inundating the target service with a flood of dummy requests, leading to the connection getting reset by over consumption of target resources. As a result, authorized and legitimate users fail to receive end to end service as requested which can lead to information as well as financial loss. Distributed Denial-of-Service (DDoS) attacks are modified version of DoS attacks which are launched from multiple IP addresses and lead to saturation of target resources. DDoS attacks are harder to detect since they originate from multiple sources, unlike DoS attacks which are triggered from a single IP address. Some of the common examples of DoS attacks are buffer overflow, ping of death (PoD), TCP SYN, smurf and teardrop attacks.

Network attack - Attackers launch network attacks with the malicious intent to compromise the network security by manipulating the network protocols being used by the layers ranging from data link layer to application layer corresponding to the OSI model. Network attacks can result in unauthorized use of network and administrative privileges, and cause damage to network resources and bandwidth. This leads to legitimate authorized users unable to gain access to the system resources and services. Packet injection and SYN flood are commonly used network attacks.

Physical attack - Malicious attackers use physical attacks such as cold boot and evil maid in order to cause physical damage and loss to the network and computer systems of the target information system making them unavailable for legitimate use.

Password attack - The intent of password attack is to gain access to legitimate user passwords which are then used to hack into the authorized systems. SQL injection attack is one of the most popular attacks of this category.

Information Gathering attack - Information gathering attack, as the name suggest, aims to gather all the relevant information related to the system vulnerabilities which can be exploited further to initiate other types of attacks on the information system. Information - related system and network vulnerabilities are collected by activities like scanning and probing. SYS scan, FIN Scan and XMAS scan are some of the examples of

information gathering attacks.

User to Root (U2R) attack - User to root attack (U2R) is usually launched for illegally obtaining the root's privileges when legally accessing a local machine. Attacker accesses the system as an authorized user; however, it later upgrades its own access privilege to super user by exploiting vulnerabilities like sniffing passwords, dictionary attack or social engineering.

Remote to Local (R2L) attack - Remote to local attack (R2L) has been widely known to be launched by an attacker by sending packets to a remote system over the network without having a valid account on that system and eventually gain unauthorized access to the entire network via the victim machine.

Probing - Probing attack is used to scan the network for valid IP addresses and gather important information such as services offered, operating system being used, and information related to host data packets.

Common Gateaway Interface (CGI) scripts - This category of attacks make use of common gateaway interface (CGI) scripts to create illegitimate and malicious inputs to web server with the intent to lure the victim to divulge important personal information such as passwords and credit card details. Phishing emails are popular examples of this attack type.

Brute Force attack - Brute force attacks are executed by attempting to illegally procure the authorized user name and password for a network system , by trying all the combinations of username and passwords that have been pre-collected by means of other attacks. Usually, this task is automated and run by a computer application which is programmed to guess the user login and passwords.

Browser based network attacks - These attacks are constructed by exploiting JavaScript and Cross-site scripting to lure the target to click on the malicious script which leads to downloading malware or directs the user to a hoax website once the link is clicked.

Shellshock attacks - This attack is executed by targeting the vulnerabilities in the command line shell which is also called BASH in Linux, UNIX and IOS operating systems. This vulnerability allows attackers to penetrate computer systems using a remote code where a string of random characters is included before the corrupt code. Bash cannot determine what action needs to be taken for these special characters and just executes the malicious code which is just after the random character string. Windows has a low chance of being exploited as the BASH vulnerability is mostly patched for Windows operating system. However, almost 80 percent of the Apache servers run on Linux, and nearly 75 percent of the internet applications are Apache based. Hence, it can be stated that all of the Internet is still highly vulnerable to shellshock attacks [52].

OpenSSL attack - OpenSSL attacks are intended to intercept encrypted communication and redirect it to another network where the data is decrypted. Once the attackers have access to the decrypted data, they use the same to gain access to the applications. Heartbleed attack is one of the most popular attacks of recent times that exploits the vulnerability in OpenSSL. Attackers use Heartbleed to deceive the victims and gain confidential user information like usernames and passwords.

Backdoor attack - Backdoor attacks are used by cybercriminals to gain illegitimate access to a legitimate website. They exploit the unsecured entry points in the system such as unpatched plug ins or input fields, to pass malware into the target system. This malware propagates through the system network and gains valuable access to the user or organization's sensitive and intellectual property, which can be highly dangerous [143].

Botnet - A botnet is a network of compromises computer systems that are being remotely controlled and operated by exploiting the system and software vulnerabilities. The term - BOT is given to each of the hacked system in the network and hence, the

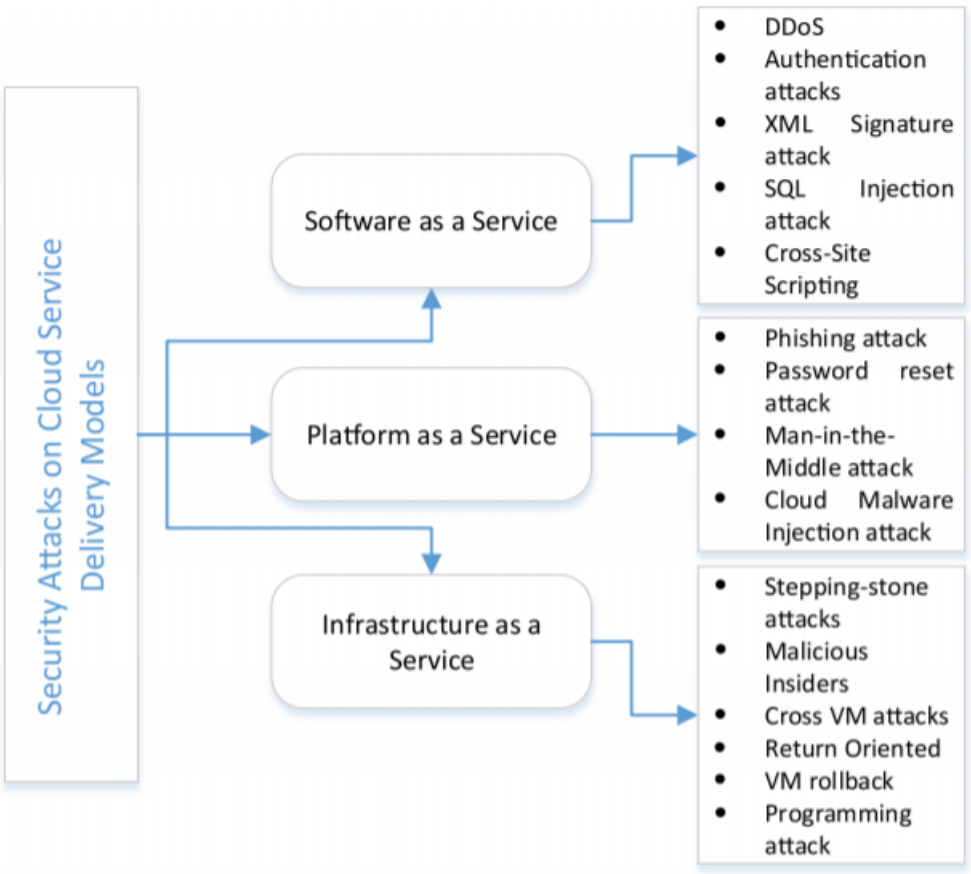

Figure 2.5: Threat profile for cloud computing [85]

name Botnet. Botnet differs from other existing malware as it makes use of Command and Control (C&C) to execute its activities [9].

Figure 2.5 lists the security threats in enterprise networks like cloud. We can see that some of the attacks in cloud such as DDoS, Man-in-the-middle and SQL injection attacks, are same as traditional network systems. However, these attacks may be conducted in distributed manner in the cloud, due its VM based topology, thus causing huge monetary and resource loss.

Hindy et al., [79] have presented an exhaustive and comparative review based on network threats and the available IDS datasets for 85 intrusion detection research papers. The research in their article covers nearly a decade of research work done in the area of intrusion detection since 2008 up until date. Figure 2.6 represents the network threats that have been considered by the authors in their paper for the last ten years. As we see in the distribution, most of the IDS research has been focused to

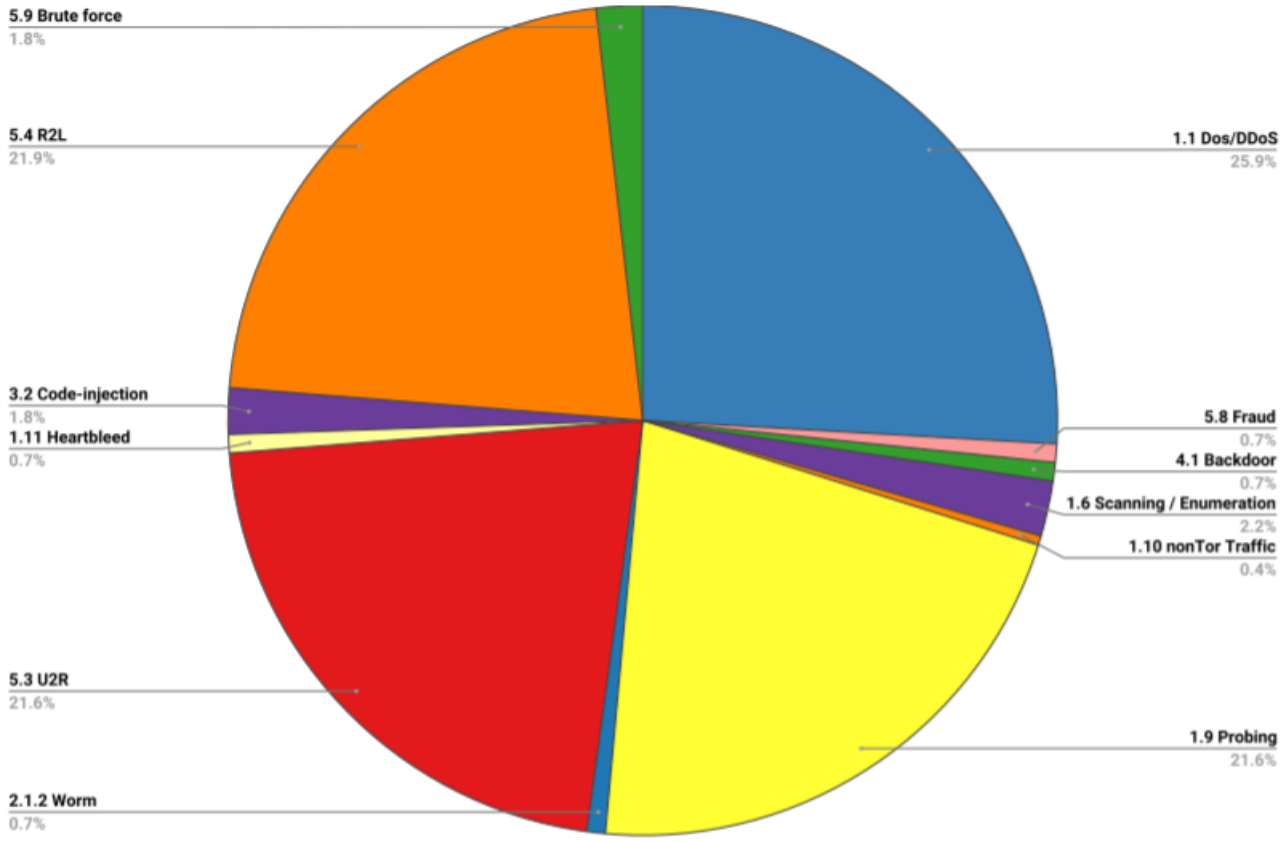

Figure 2.6: Distribution of network threats covered in [79]

counter DoS/DDoS, R2L, U2R and probing attacks and very few modern attacks such as Heartbleed, Brute force and SQL code injection have been addressed by the covered IDS schemes. We can conclude that this may be due to the unavailability of a contemporary IDS dataset until 2015 when UNSW - NB15 dataset was published, followed by CICIDS - 2017 dataset. These datasets cover the more sophisticated attacks from our current time. However, much research is done primarily on DoS and DDoS attack categories in the past which may not provide a comprehensive defense strategy for IDS based solution. Hence, based on our exhaustive literature review on network threat taxonomy in this section, we conclude that it is important to consider all attack classes available with the latest CICIDS - 2017 dataset for implementation and analysis of our proposed IDS framework.

2.3 Review of Intrusion Detection Datasets

The publicly available data sets which researchers have been using to implement and evaluate their intrusion detection schemes include DARPA, KDD'99, DEFCON, CAIDA, LNML, CDX, Kyoto, Twente, UMASS, ISCX2012, ADFA, UNSW-NB15 and CICIDS 2017. We give a short description for these datasets below:

(1) DARPA (Defense Advanced Research Project Agency 1998-1999) Dataset: DARPA is a publicly available dataset developed by MIT Lincoln Laboratory. This dataset is one of the first IDS dataset that has been extensively used by researchers since 1998 for IDS testing and validation, This dataset was generated by simulating network traffic similar to an Air force base and consists of benign and attack records with training data for seven weeks and testing data for two weeks. Simulation of various attacks such as DoS, Guess password, buffer overlfow, remote FTP, Syn flood, port scans, and rootkits in the DARPA dataset are based on network activities that include e-mail, browsing, FTP, Telnet, IRC and SNMP [51] [153].

This dataset has been heavily criticised by academic researchers, for not being relevant for the evaluation of IDSs and considered as outdated in current times. This is due to the lack of representation of modern day attacks and also the current network infrastructure [153]. Another criticism of this dataset is the high redundancy in the records and the use of artificial attack injection [141].

(2) KDD'99 (1998-1999) Dataset : KDD '99 dataset is derived from DARPA 98 by converting the tcpdump into fundamental attributes such as the count for unsuccessful login attempts [141]. This dataset consists of 41 features and provides two types of training data - full training data set and 10% training dataset and has been quite popular with IDS testing in the absence of any better benchmark IDS datasets.

KDD 99 dataset includes records for 22 attack types. Further to this, the attacks can be divided broadly into 4 classes which are Denial-of-Service (DoS) attacks, U2R, R2L and probe attacks [26]. However Ttavallaee et al. [174] highlighted several drawbacks and limitations of KDD 99 dataset such as the the merging of benign and attack TCP network traffic, data record redundancy, dropped data packets leading to loss of data and hence, leading to a biased output result in IDS testing experiments.

(3) NSL-KDD Dataset : NSL-KDD Dataset was published in 2009, by Tavallaee et al. [174] with the sole aim to address the issues presented by KDD 99 dataset. The redundant data records from KDD 99 were removed and the data was further cleaned to ensure that the training and testing data records were of an acceptable count of 257673 records. However, just like KDD 99 data, each record consists of the same 41 features and a label class to identify a normal record or an attack.

Since the training and evaluation data records is not a huge number, IDS testing and evaluation can be done on the complete data set instead of selecting random smaller data parts. NSL-KDD dataset comes with the advantages of having no redundant data records in the training set as well as no duplicate records in the evaluation set, thus making it a widely used dataset for IDS even today [25] [26].

(4) DEFCON (Shmoo Group 2000-2002) Dataset : The DEFCON dataset is similar to data repository and was created by capturing the network traffic as a part of a hacking competition by the name of Capture the Flag (CTF) [69]. This traffic mainly consisted of attack packets from port scan, sweeps, intruding administrative privilege via unauthorized access and FTP by telnet protocol attacks [153]. Since this dataset does not consist of any normal traffic and only intrusive traffic is included, it is used for evaluation of alert correlation techniques [25].

(5) CAIDA (Centre for Applied Internet Data Analysis 2007) Dataset : CAIDA data repository collects numerous types of data and provides the same to researchers. Most of the CAIDA datasets are specific to a particular attack event with the payload, protocol information and destination details being made anonymous [78].

(6) LBNL (Lawrence Berkeley National Laboratory 2004-2005) Dataset : Network traces in LBNL dataset is full header network traffic and does not consist of any payload. Heavy anonymization is done on this dataset inorder to remove any information that may lead to identification of the individual IP addresses [124].

(7) CDX (Cyber Defense Exercise 2009) Dataset : Sangster et al. [148] suggested to use network warfare competitions and use them to generate a labelled dataset for current times. Web, email and DNS lookups are included in the captured network traffic and activities like reconnaissance and automated attack launch is achieved using tools like Nikto, Nessus and WebScarab. However this dataset is not quite diverse and does

not have a reasonable record number for IDS testing. It is preferably used for testing IDS alert rules [153].

(8) Kyoto (Kyoto University 2009) Dataset : Kyoto dataset is easily available to the public and has been generated using honeypots and hence, only those attacks that were directed at the honeypots were observed. Hence, it does not represent diverse attack scenarios. Normal traffic simulation included only DNS and email network traffic, hence, this dataset does not represent real life network traffic [159].

(9) Twente (University of Twente 2009) Dataset : This dataset has been created using Netflow to capture data from a honeypot network. OpenSSH, Apache web server and Proftp services using Ident authentication on port 113, are included in this dataset. However, one of the drawbacks of this dataset is that it is small and does not include diverse attacks [22].

(10) UNMASS : UNMassTraceRepository [99] supplies researchers several network traffic traces as well as traces from wireless application. Some traces have been given voluntarily while others have been collected by the dataset archive suppliers. Currently there are 19 packet - based data sets available from UNMASS repository [141]. However the network traffic and attacks included in these traces are not diverse and hence, this dataset is not used for testing IDS and IPS methods [153] .

(11) ISCX2012 (University of New Brunswick 2012) Dataset : ISCX2012 dataset is a labelled dataset and consists of a wide range of attacks. It is generated by observing and analysing traces in the real time network traffic of HTTP, SMTP, SSH, IMAP, POP3, and FTP protocols and using the same to recognize normal activity in the computer network [155].

(12) UNSW-NB15 (University of New South Wales 2015) Dataset : UNSW-NB15 dataset has been proposed by Dr. Nour Moustafa [118]. This dataset has been generated using the IXIA traffic generator and consists of pcap files. The records include benign traffic and attack classes such as Fuzzers, Analysis, Backdoor attacks, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms. The issue with this dataset is that the number of attack records described as "generic" is quite large and this leads to ambiguity [42].

(13) ADFA (University of New South Wales 2013) Dataset : Australian Defense Force Academy (ADFA) and University of New South Wales collaboratively published the ADFA dataset in 2013 [48]. There are essentially two datasets : ADFA-LD and ADFA-WD which together are commonly referred to as ADFA dataset in literature. This dataset was generated using system call based HIDS and consists of Linux operating system calls in ADFA-LD and Windows operating system calls in ADFA-WD.

ADFA-LD includes system call traces of various attacks. Various penetration testing tools are used along with latest attack techniques to exploit the preset vulnerabilities of the system. ADFA-LD consists of 833 benign system call traces for training, 4372 traces to identify and examine false-alarm rate, and 746 traces which enlist six different attacks for testing purpose. This dataset only provides the list of system call numbers [101].

(14) CICIDS-2017 (University of New Brunswick 2017-2018) Dataset : University of New Burnswick has published a new intrusion detection dataset - CICIDS 2017 which is created by capturing complete traffic using 12 different machines set up within the Victim Network and attacks are launched from the Attack Network. CICIDS2017 dataset includes all available protocols like http, https, SSH, FTP and SMTP. This dataset has been generated taking into consideration the latest and most relevant attack scenarios like DoS, DDoS, Brute Force, Web Brute force, XSS, SQL injection, Heartbleed attack, slowloris, slowhttptest, Hulk DDoS attack, GoldenEye attack, infiltration, Port scan and Botnet [153].

This new dataset addresses important evaluation criterion such as complete network configuration, complete traffic, labelled dataset, complete interaction, complete capture, available protocols, attack diversity, heterogeneity, feature set and meta data [153]. CICIDS-2017 dataset is more relevant in current times as it is generated with the comprehensive network set up. Further, the completeness and diversity of attack classification provided by the CICIDS ‚Äì 2017 dataset features provide interesting research scope for developing feature selection and intrusion detection model. Hence, we have chosen to work with CICIDS-2017 dataset for evaluating our proposed ensemble based network intrusion detection model. We will explain the steps and analysis of the dataset in detail in Chapter 5 of our thesis.

Dataset	Realistic Traffic	Label data	IoT traces	Zero-day attacks	Full packet captured	Year
DARPA 98	✓	✓	✖	✖	✓	1998
KDDCUP 99	✓	✓	✖	✖	✓	1999
CAIDA	✓	✖	✖	✖	✖	2007
NSL-KDD	✓	✓	✖	✖	✓	2009
ISCX 2012	✓	✓	✖	✖	✓	2012
ADFA-WD	✓	✓	✖	✓	✓	2014
ADFA-LD	✓	✓	✖	✓	✓	2014
CICIDS2017	✓	✓	✖	✓	✓	2017
Bot-IoT	✓	✓	✓	✓	✓	2018

Figure 2.7: Comparison of selected benchmark IDS Datasets [90]

Khraisat et al., [90] have given a brief comparative summary for a few of the datasets described above and this is shown in Figure 2.7. Figure 2.8, presents the ratio of how the publicly available IDS datasets have been used in IDS literature since 2008 [79]. It is noteworthy to mention that during the last decade, nearly 50 percent IDS research papers (out of 85 that are reported) used KDD 99 dataset which was followed by NSL-KDD dataset, even though these datasets are outdated and do not include the sophisticated attacks and network topology of current times. We also make an observation that the recently published, modern IDS dataset CICIDS - 2017 is still not used for IDS performance evaluation as it it noted that only 2 percent worked with CICIDS - 2017 dataset. Hence, CICIDS - 2017 becomes an obvious choice to work with in our research work which can further reinforce it to be used as a benchmark dataset for future IDS research performance evaluation.

2.4 Literature review of related works

We have briefly discussed several approaches to implement intrusion detection systems in section 2.1.6. Building IDSs with the aim of catering to modern times is immensely challenging considering the world is running on high internet speeds and huge volumes of network data are being exchanged due to enhanced computer activities. This is one of the biggest challenges faced while designing any network intrusion detection systems in modern times which are used to continuously monitor the network logs and network traffic [64]. Enterprise networks such as Cloud computing have introduced a new area of network and system security. Cloud computing has a new set of security challenges due to its unique characteristics and hence, the IDS solution for the cloud needs to ad-

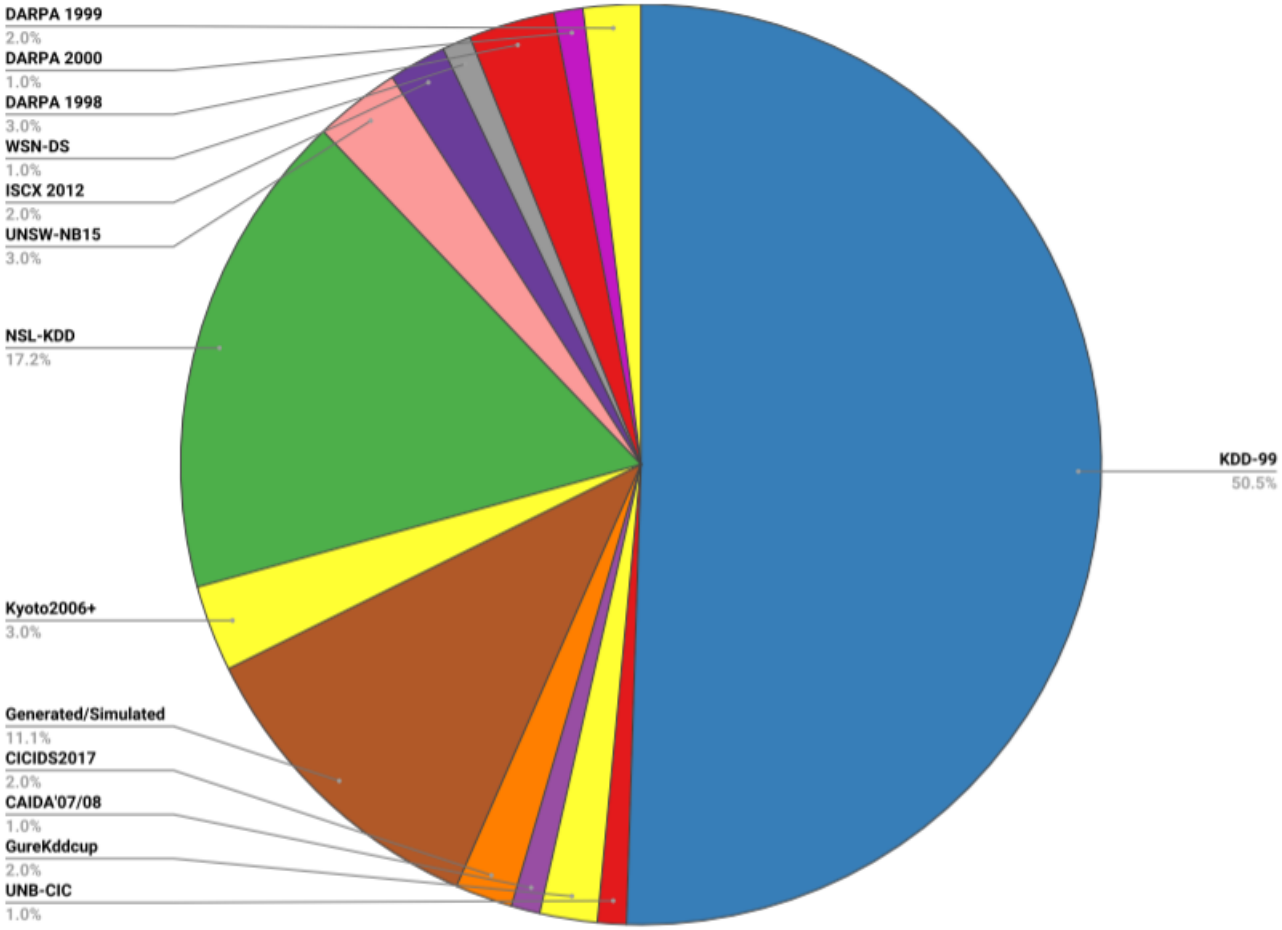

Figure 2.8: Distribution of datasets for evaluating IDS performance in literature [79]

dress these while designing IDS framework for cloud infrastructure. Figure 2.9 broadly describes the cloud computing security issues [85].

As seen above, cloud has a separate level of threat vulnerability, given that there are more than one points of entry. This makes it easier for sophisticated attacks to be conducted in a distributed manner through more than one entry point. [173] have tried to address the need for intrusion detection in cloud by proposing a collaborative IDS framework as shown in Figure 2.10. With this approach, the authors try to point out that enterprise networks like cloud need Collaborative IDS methods instead of standalone IDSs as CIDS are better at identifying cooperative attacks. Their approach uses cooperative nodes and a central coordinator. The cooperative nodes have both HIDS and NIDS deployed on them, depending on which location they are monitoring traffic. If an intrusion is detected, these agents report it to the central coordinator, which is

Security Issues	Attack vectors	Attacks Types	Impacts
Virtualization level Security Issues	• Social engineering • Storage vulnerabilities • Datacenter vulnerabilities and Network • VM vulnerabilities, etc.	• DoS and DDoS • VM Escape • Hypervisor Rootkit	• Software interruption and modification (deletion) • Programming flaws
Application level Security Issues	• Session management and broken authentication • Security misconfiguration, etc.	• SQL injection attacks • Cross Site scripting and • Other application based attacks.	• Modification of data at rest and in transit • Confidentiality • Session hijacking
Network level Security Issues	• Firewall misconfiguration, etc.	• DNS attacks • Sniffer attacks • Issues of reuse IP address • Network Sniffing, VoIP related attacks (e.g. VoIP phishing).	• Traffic flow analysis • Exposure in network
Physical level Security Issues	• Loss of Power and environmental control	• Phishing Attacks • Malware injection attack	• Limited access to data centers • Hardware modification and theft

Figure 2.9: Cloud computing security issues [85]

capable of generating a bigger attack picture, thus capturing any advanced distributed attacks on the cloud network.

A collaborative intrusion detection framework shown in Figure 2.11, suggests using an open source NIDS tool called SNORT [157] and open source HIDS tool namely OSSEC [131] as detection mechanisms on multiple nodes in a public cloud network. SIEM (Security Information and Event Management) or SEIM (Security Event and Information management) [164] tools such as SPLUNK are used as the central coordinator. The alerts generated by the HIDS and NIDS are registered with the central coordinator unit and the SEIM generates attack detection patterns and visualisations using alert correlation to assist the network administrator with intrusion detection.

Several researchers have proposed data mining and machine learning techniques for IDS implementation as a solution to overcome the challenges associated with high computational time for NIDS while processing a huge amount of data and also to achieve high detection accuracy while reducing the false alarm rates in NIDS [132]. Machine learning is used as a learning procedure to gather information from various datasets. This extracted knowledge is used via computing based resources in applying rules and complex functions to build mathematical and scientific models , which are further used in pattern recognition tasks or for prediction of intrusive behaviour [57]. The main aim of applying machine learning based approaches in IDS design , is to reduce human knowledge involvement and increase the IDS efficiency.

Machine learning (ML) based approaches for intrusion detection can be divided into different categories. Zamani et al. [190] classify ML based IDS into Artificial Intelligence

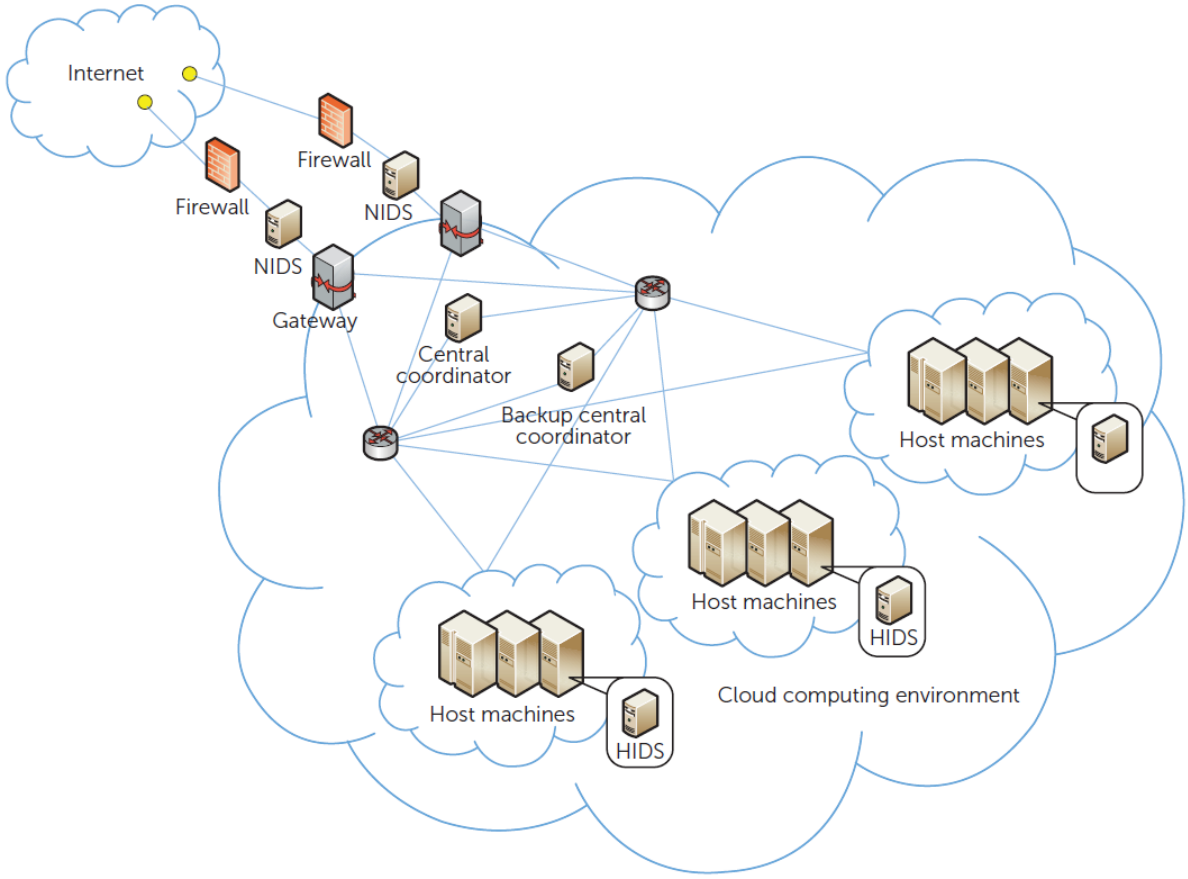

Figure 2.10: Cloud computing security issues [173]

(AI) based techniques and Computational Intelligence (CI) based techniques. AI based approaches include traditional methods using statistical modelling. The authors mention k-Nearest Neighbor (KNN), Multi-layer Perceptron (MLP) and Support Vector Machines (SVM) as AI based techniques used in IDS implementation. On the other hand, CI based approaches are built to solve issues that are beyond the scope of AI based approaches. CI based methods are inspired by nature and include evolutionary computing, fuzzy logic, artificial neural networks and artificial immune systems. Mishra et al. [111] classify the machine learning methods based on the detection method used . Hence, their work focuses mainly on approaches using signature based detection, anomaly detection or hybrid detection methods and also considers feature selection methods used for data processing. Using this division, they classify the ML based IDS approaches: (1) Single classifiers with all features of the data set (2) Single classifiers with limited features of the data set (3) Multiple classifiers with all features of the data set and (4) Multiple classifiers with limited features of the data set. A single classifier model involves only

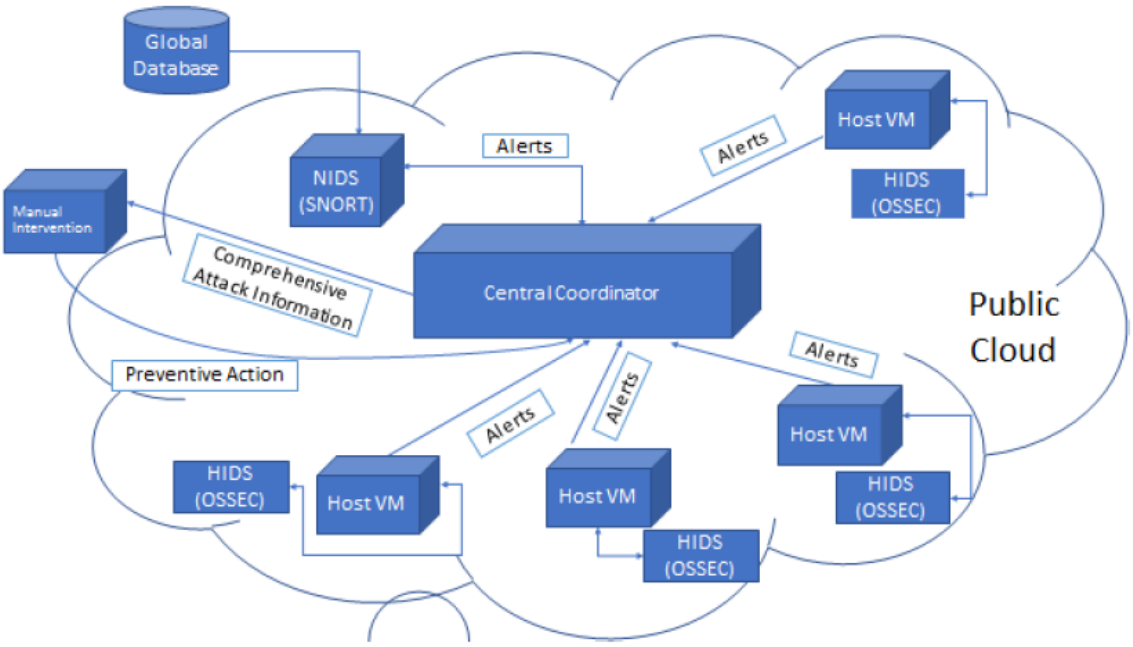

Figure 2.11: Collaborative Intrusion Detection Scheme for Public Cloud protect[122]

one classifier for intrusion detection. Multiple classifier model involves two or more classifiers whose output is further integrated to generate a common result for intrusion detection [111].

Further machine learning algorithms are also categorized as supervised ML techniques and unsupervised ML techniques. Supervised ML based IDS approach involves a training stage in which the ML algorithm learns and builds a normal profile of the system by training the ML classifier to learn the association between the input data and the corresponding output label which can be either normal or anomaly. Hence, this approach works with labelled training data set. In the testing phase, the trained classifier model is used to predict the class for any given set of training data [90].

2.4.1 Supervised Machine Learning Algorithms

We describe some of the popular supervised machine learning techniques below.

1. Decision Trees (DT) : Decision trees are used to represent all possible results for a decision by using a branching technique. There are three elemental parts for a decision tree. Decision node is the first main unit and is used to define the test characteristic. Next component is the branch of the decision tree, which represents a potential decision depending on the test value. The third main component is the leaf which defines the class to which the test instance corresponds [146].

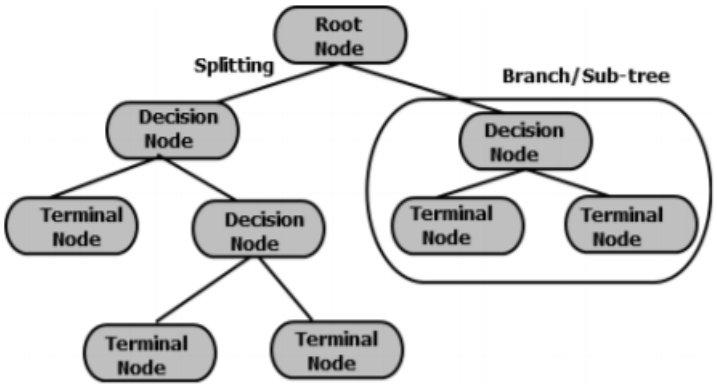

Figure 2.12: Decision Tree representation [111]

Figure 2.12 gives us the graphical representation of the decision tree. Some prominent decision tree algorithms include C4.5/J48, ID3, CART.

2. Naive Bayes (NB) : Naive Bayes is a classification machine learning algorithm that is statistical in nature. It uses the Bayes theorem [35] to predict the class probability. Naive Bayes approach assumes that the class attributes are independent of each other [163].

3. Genetic Algorithms (GA) : Genetic algorithms are machine learning methods that are based on the principles of evolution by Charles Darwin which explains the process of natural selection by which only the fittest of individuals are selected for progeny. As shown in figure 2.13, genetic algorithms involve five main components : initial population, fitness function, selection, crossover, and mutation. As shown in the execution flow, GA algorithm starts with the identification of initial population, each member of the population is considered as a solution and is characterized by a fixed number of genes (which can be defined as features) to form a chromosome (solution to the search problem). The fitness function provides a fitness score to each individual based on which they may

be selected for reproduction. During selection phase, two parents are selected based on the fitness score and are chosen for progeny. Crossover is a crucial phase and the crossover point is randomly selected. As the parent genes are exchanged, new offspring are generated and are further added to the initial population until the crossover point is reached. The process terminates once the best possible solution is reached [111].

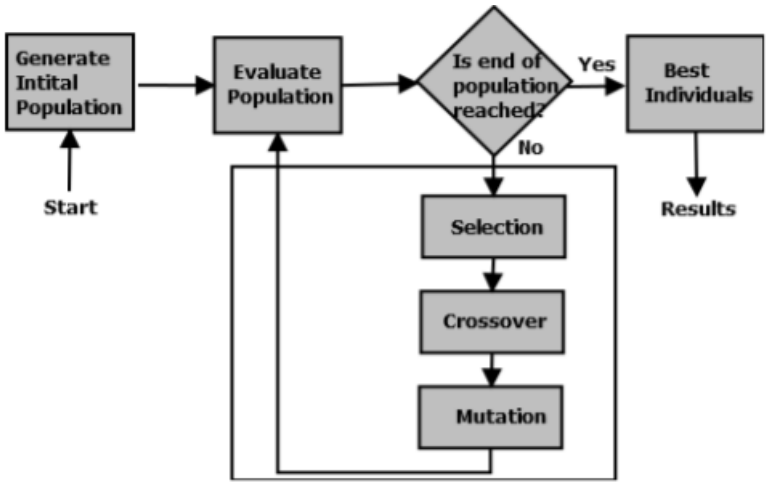

Figure 2.13: Generic Algorithm flow of execution [111]

4. Artificial Neural Network (ANN) : Artificial Neural Network (ANN) is one of the most robust and widely used machine learning technique. ANN consists of three different layers: Input layer, Hidden layer and Output layer as shown in Figure 2.14. The elemental units in neural networks are the nodes. Neural networks can be based on multi layered perceptrons with Back propagation, Radial Basis, Adaptive Resonance theory, Hopfields Networks and Neural tree [111]. These algorithms work with input information in two ways : Feed Forward networks and Back propagation. In Feed forward propagation, the input is fed via the input layer to every node in the hidden layer which calculates the weight and transforms the result to the output layer. Error value is calculated by measuring the difference between the actual value and the desired value. The Back propagation (BP) method differs from the feed forward network by the manner in which the error is handled. In BP, the error is transferred back to the input layer from the output layer and the weights are re-adjusted until a pre - defined threshold value is reached [27].

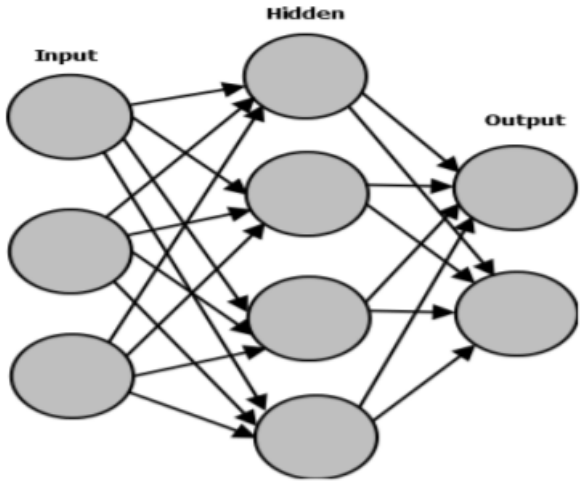

Figure 2.14: Various layers in Neural Network [111]

5. Support Vector Machine (SVM) : Support Vector Machine (SVM) is an extensively used machine learning method which aims to define a hyperplane with the maximum margin that can classify the two data classes distinctly. The data points that are closest to the hyperplane are called support vectors. The choice of these support vectors defines the location and orientation of the hyperplane. The dimensionality of this hyperplane is based on the number of input features, The hyperplane is a straight line when there are only two input features as seen in Figure 2.14. When there are three input features, it is a two- dimensional plane. However, there may be a possibility that the input dataset

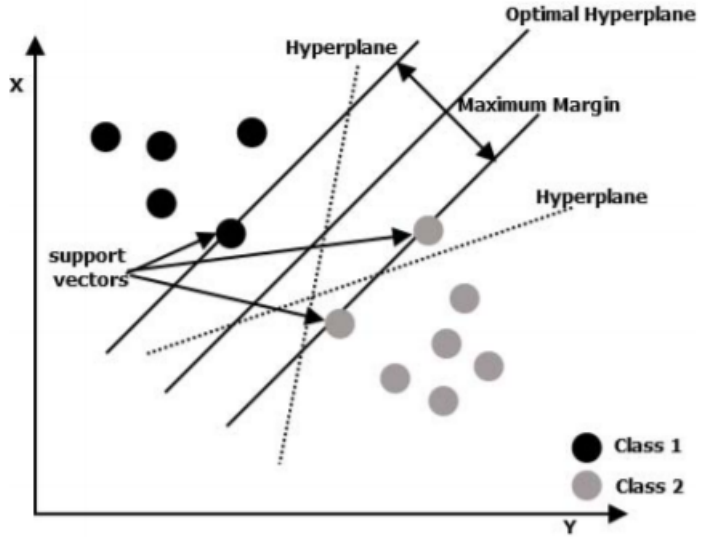

Figure 2.15: Linear SVM [111]

consists of non-linear features and hence, cannot be linearly separated by a straight line. The solution to this problem is achieved by mapping the data to a higher dimensional feature space where it is easy to identify the hyperplane for feature classification. Kernel functions include radial basis kernel (RBF), Polynomial Kernel, which are most commonly used for SVM. Based on type of Kernel functions used, SVM can be classified as Linear SVM and Non-linear SVMs [111].

6. Hidden Markov Model (HMM) : Hidden Markov Models (HMM) are based on Markov models and are an extension of Markov chains. Both these algorithms are classified under Markov Models. Markov chains are use to predict the probability of events and are based on the previous state assumption. However, HMM is used for systems that have unknown or hidden Markov chains or processes. Veterbi- algorithm is used to decode the hidden states of HMM [111].

2.4.2 Unsupervised Machine Learning Algorithms

On the other hand, unsupervised learning methods are used with datasets with no labels. They gather relevant knowledge using random variables from the input data and generating a collective density model for the given dataset. Unlike supervised learning models where the output class is already available, unsupervised ML learning methods classify the data into class groups automatically during the learning phase. The logic used to identify normal clusters and intrusion clusters is based on the fact that the majority of the normal instances will form the biggest cluster. On the contrary, the intrusions or anomaly instances will be smaller as compared to the normal data and hence, will form much smaller clusters comparatively. Unsupervised learning methods are beyond the scope of the research work presented in the research presented in this thesis.

The manuscript by Hindy et al., gives an overview of the algorithms used by intrusion detection schemes during the last ten years. This is based on the 85 research papers analysed by them. Figure 2.16 depicts an abridged description of the various IDS methods used up until now and the importance of machine learning techniques for network intrusion detection is clearly evident [79].

2.4.3 Ensemble based Intrusion Detection system

Dietterich explains how ensemble methods work as compared to ordinary machine learning algorithms in his paper [56]. Every data point in space is made up of feature vectors (represented by x) and a class label y with an assumption that each data point (x,y) is defined by a function f such that $y = f(\mathrm{x})$. The main aim of the machine learning algorithm is to find the nearest best possible guess h for y such that h can be implemented to define the labels y for new values of x. Here, h is termed as the *classifier*. Commonly used machine learning algorithms aim to find the best possible function h for the function f by searching through all possible functions which are called *hypothesis*. The best possible h is decided by the criteria as to how closely can h map f for the training data points. Contrary to the traditional machine learning algorithms, ensemble techniques work on developing a set or committee or ensemble of *hypothesis* and use *majority vote* to predict the class label for unknown data points, rather than looking for a single optimum hypothesis.

Figure 2.16: Distribution of Intrusion Detection Techniques based on research included in [79]

Ensemble based approaches have proven to be accurate and flexible as compared to traditional machine learning approaches as these methods can be used for improving the detection rate and accuracy of a weak learner classifier. In the age of cloud and distributed computing, intrusion detection systems designed using ensemble techniques can easily be implemented and are capable of handling the increased computational load without compromising the performance of the IDS [92] [64]. The most widely used

algorithms for ensemble techniques are bagging, boosting, majority voting and stacking [90]. Bagging method employs training a chosen classifier on various subsets of the same dataset. It is also popularly called Bootstrap aggregation. This technique is used to reduce the variance for a given ML method [111]. Boosting is an iterative technique which uses random, non repeated training data samples to train the weak classifier in the first iteration and keeps repeating the iteration until a strong classifier has been generated from the base weak classifier [65]. Stacking method is used to merge the predictions generated by n base classifiers for the chosen dataset using a meta algorithm and provide a ultimate prediction [111]. We chose to use ensemble based technique to propose a novel network intrusion detection system which is elaborated on in Chapter 4 of this thesis.

2.5 Commonly used metrics employed for IDS evaluation

Several performance evaluation metrics have been listed in IDS literature and have been used by researchers to validate their IDS schemes for anomaly detection. We describe the common terms used to describe the performance of classification schemes below [188].

1. *True Positive (TP) :-* The number of dataset instances that are correctly identified as Benign or Normal class.
2. *True Negative (TN) :-* The number of dataset instances that are correctly identified as Intrusion or Attack class.
3. *False Positive (FP) :-* The number of dataset instances that are incorrectly identified as Intrusion or Attack class.
4. *False Negative (FN) :-* The number of dataset instances that are incorrectly identified as Benign or Normal class.

The above mentioned common terminology for IDS detection is further used extensively in advance evaluation metrics which we enlist below and briefly describe each one of them.

1. *Accuracy (Acc) :* Accuracy is the percentage of correctly classified or predicted records in a testing data [147]. This includes the correctly predicted true positives and true negatives. Accuracy is calculated using the below formula :

$$Acc = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{2.1}$$

2. *True Positive Rate (TPR) :* True positive rate is the number of malicious instances that were correctly classified as intrusions. It is also called detection rate or sensitivity or *recall* [147] and is calculated using the below formula :

$$TPR = \frac{TP}{(TP + FN)} \tag{2.2}$$

3. *False Positive Rate (FPR) :* False positive rate is also called false alarm rate or fall-out [147]. It indicates the number of records that were benign and were incorrectly classified as attacks. It is calculated using the below formula :

$$TNR = \frac{FP}{(FP + TN)} \tag{2.3}$$

4. *True Negative Rate (TNR) :* True negative rate is the number of attack patterns that were correctly predicted as intrusions. It is also know as specificity or selectivity and is calculated using the below formula :

$$TNR = \frac{TN}{(TN + FP)} \tag{2.4}$$

which is equivalent to

$$TNR = 1 - FPR \tag{2.5}$$

5. *False Negative Rate (FNR) :* False negative rate is the proportion of benign records that were incorrectly classified as attacks and is calculated using the below formula:

$$FNR = 1 - TPR \tag{2.6}$$

6. *Precision :* Precision is also known as Positive Predictive Value (PPV) and is the measure of the correctly classified instances for both benign and attack patterns. It is calculated by the below formula :

$$Precision = \frac{TP}{(TP + FP)} \tag{2.7}$$

7. Negative Predictive Value (NPV) : Negative predictive value is the measure of correctly predicted attack instances which are classified as attacks in real data and is calculated by the below formula :

$$Negative\,Predictive\,Value = \frac{TN}{(TN + FN)} \tag{2.8}$$

8. *F-measure :* It is also called *F-score* or *F-value* and it is the harmonic mean of *recall* and *precision*. F-measure is calculated using the below formula :

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.9}$$

9. *Matthews Correlation Coefficient (MCC) :* MCC , also know as *phi coefficient* is another important metric used to evaluate IDS model performance. It is used to measure the correlation between the projected results and the actual data. Value for MCC ranges from +1 to -1, where +1 value indicates that the detection was accurate [165]. MCC is calculated using the below formula :

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{2.10}$$

10. *ROC Curve (ROC) :* ROC curve (*Receiver Operating Characteristic curve*) is generated by plotting True Positive Rate and False Positive rate. ROC shows the performance of the IDS model at various thresholds. ROC can be further interpreted by AUC (Area under the ROC Curve). AUC is the total area under the ROC curve and its value ranges between 0 and 1 [71].

11. *Confusion Matrix :* Confusion Matrix is also known as error matrix which provides a visualisation of the performance of an IDS anomaly prediction model. It consists of rows and columns and each row represents the actual class while the column represents the predicted class. It represents the number of True Positives, False positives, False Negatives, and True negatives as shown in Figure 2.17. It is a very useful tool for calculating the advanced classification metrics described previously. We use *Accuracy, True Positive Rate, False Positive Rate, Precision, Recall, F-Measure, MCC* and *ROC Area* to validate the performance of our proposed ensemble based IDS framework.

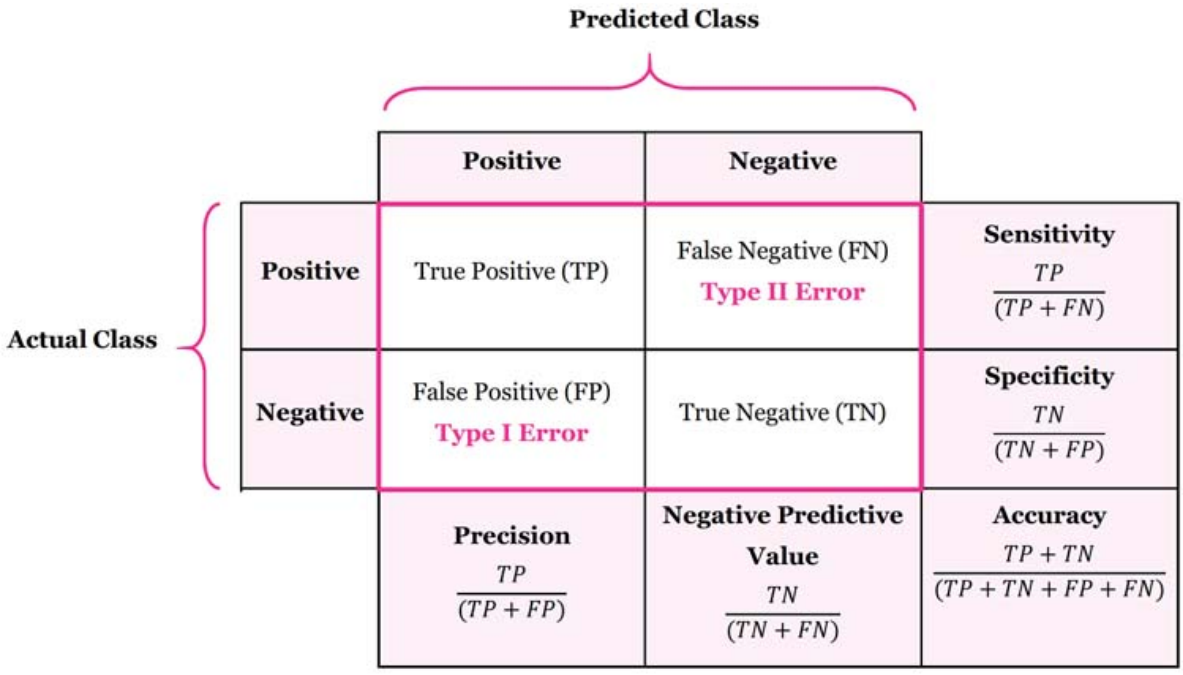

Figure 2.17: Confusion Matrix explained [156]

2.6 Chapter summary

In this chapter, we have given an elaborate explanation of the importance of intrusion detection systems and have discussed in detail the taxonomy of intrusion detection systems. In addition, we have also described the traditional and latest sophisticated network threats which require urgent risk management for modern network systems. We have also provided a brief review of all the publicly available intrusion detection datasets with the advantages and disadvantages of each. A literature review is provided on the significance of machine learning classifiers for intrusion detection and anomaly classification problems. Further, ensemble based intrusion detection methods in literature have also been discussed in brief in this chapter. The commonly used metrics used for evaluation of IDS schemes are also explained in this chapter. We have provided a detailed literature review on ensemble based intrusion detection research in Chapter 4.

CHAPTER 3

PROPOSED ENSEMBLE BASED FEATURE SELECTION APPROACH

The role of intrusion detection is critical when it comes to security of network systems as it assists network security managers to identify and take preventive actions against malicious activities such as network attacks. Hence, despite major network security advancements, it is understandable why the research community is focused on trying to analyse novel intrusion detection schemes for robust network security solutions. The traditional intrusion detection datasets such as DARPA (Defense Advanced Research Projects Agency) and KDD (Knowledge Discovery and Data mining) were made available in 1999. Even though they are heavily criticized by researchers [108], they are still being used due to the lack of modern bench marked dataset. Section 3.1 continues with IDS dataset description provided in chapter 2, and describes different attack categories and data set features in much detail for the KDD99, NSL-KDD and CICIDS - 2017 datasets. It also provided the test bed framework, data set record distribution and dataset feature information for CICIDS - 2017 dataset.

Section 3.2 introduces feature selection methods using machine learning techniques and provides a discussion on why ensemble-based methods are needed to implement feature selection for high dimensionality datasets. Our proposed ensemble based feature selection approach is presented in section 3.3 and chapter summary is provided in section 3.4.

3.1 Further discussion on KDD 99, NSL-KDD and CICIDS-2017 datasets

Research [125] shows that nearly 50 percent of the intrusion detection studies use these two datasets for testing their works. It is interesting to note here, that these data sets were published more than two decades ago and thus represent the network conditions and security threats from those times. These datasets lack generalisation and representation of the latest network security trends and hence, it would be very misleading to test modern day IDS frameworks using these bygone datasets as the IDS models trained using these datasets would not give the true snapshot of network security in current times [48] [174]. Another issue in publishing IDS datasets publicly is the need for anonymity of the data and hence, may not reflect the threat scenarios and it may have weak features due to concerns for security [175].

In order to address the aforementioned issues with traditional intrusion detection datasets, Tavallaee et al. [174], proposed NSL-KDD which addresses the drawbacks for KDD 99 such as data redundancy, irrelevant features and imbalanced sampling. NSL-KDD is comparatively more balanced than KDD but still contains skewed data due to the minority classes from KDD 99. It also suffers from the inherent aging issue similar to KDD 99. Section 2.3 of chapter 2 enlists the publicly available IDS datasets. These were created over several years by several researchers who aimed to develop new IDS datasets to meet the testing requirements when it came to validation of IDS schemes. Many of the datasets may satisfy partial criteria for IDS dataset characteristics as they may focus on specific threats or may have been created synthetically to meet the needs of particular research [147]. Table 3.1 provides a comprehensive summary of the publicly available datasets with the details of the features and attack types included in each one of them [176].

Dataset Name	**Developed By**	**Total Features**	**Attack Category**
DARPA	MIT Lincoln Laboratory	41	Dos, R2L, U2R, Probe
KDD 99	University of California	41	Dos, R2L, U2R, Probe
NSL-KDD	University of California	41	Dos, R2L, U2R, Probe
DEFCON	Shmoo Group	Flag traces	Telnet Protocol Attacks
CAIDA	Center of Applied Internet Data Analysis	20	DDoS
LBNL	Lawrence Berkeley National Laboratory	Internet traces	Intrusive traces
CDX	United States Military Academy	5	Buffer Overflow
Kyoto	Kyoto University	24	Benign and Attack sessions
Twente	Twente University	IP Flows	Intrusive traffic, Side-effect traffic, Unknown traffic, and Uncorrelated alerts
ISCX2012	University of New Brunswick	IP Flows	DoS, DDoS, Brute-force, Infiltration
ADFA	University of New South Wales	System call traces	Zero-day attacks, Stealth attack, C100 Webshell attack
UNSW - NB15	Australian Centre for Cyber Security(ACCS)	49	Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shell code, Worms
CICIDS - 2017	University of New Brunswick	80	Brute force, Portscan, Botnet, DoS, DDoS, Web, Infiltration, Heartbleed

Table 3.1: Comprehensive Summary of Publicly available IDS Datasets [176]

In our research work, we have used the most recent dataset called CICIDS-2017 which is offered by the University of New Burnswick using our proposed IDS framework. This dataset covers the modern attacks and also represents the network setups which are more relevant to our current times. We have also used NSL-KDD [174] to compare

the performance of the results obtained on CICIDS-2017 dataset using our framework. A detailed description of dataset features, attack categories, record distributions and test bed set up for the datasets used in our research is discussed as follows.

3.1.1 KDD 99

KDD 99 dataset was generated by the University of California by processing the tcpdump part from the DARPA 98 dataset. It is hence, an upgraded edition of DARPA 98 created by gathering tcpdump data over a span of 10 weeks. It is comprised of a training dataset and a testing dataset with a choice of a complete dataset and a 10 percent ratio of the complete dataset. This dataset consists of 41 features and five attack categories including DoS, Probe, R2L, U2R and normal. The features can be classified into basic features (feature numbers 1 to 9), content features (feature numbers 10 to 22) and time-based traffic features (feature numbers 23 to 41) respectively. Basic features were extracted using the packet headers, TCP and UDP packets that were gathered from the packet capture (Pcap) files of the tcpdump. The content features were extracted from the TCP packet payloads as it would contain attack information related to R2l and U2R attacks. Further, time-based traffic features were collected by categorizing the connection for common host and common service for a particular time window of 2 seconds [188]. Figure 3.1 lists the 41 features of KDD 99 dataset and Figure 3.2 shows the data record distributions for KDD 99 dataset.

Attack category	Description	Data instances - 10 % data			
		KDDCup 99		NSL-KDD	
		Train	Test	Train	Test
Normal	Normal connection records	97,278	60,593	67,343	9,710
DoS	Attacker aims at making network resources down	391,458	229,853	45,927	7,458
Probe	Obtaining detailed statistics of system and network configuration details	4,107	4,166	11,656	2,422
R2L	Illegal access from remote computer	1,126	16,189	995	2,887
U2R	Obtaining the root or super-user access on a particular computer	52	228	52	67
Total		**494,021**	**311,029**	**125,973**	**22,544**

Figure 3.2: Data set record distribution KDD 99 and NSL-KDD [188]

Label	Network data feature	Label	Network data feature	Label	Network data feature	Label	Network data feature
A	duration	L	Logged in	W	count	AH	dst_host_same_srv_rate
B	protocol-type	M	num_ comprised	X	srv_count	AI	dst_host_diff_srv_rate
C	service	N	root_shell	Y	serror_rate	AJ	dst_host_same_src_port_rate
D	flag	O	Stu attempted	Z	srv_serror_rate	AK	dst_host_srv_diff_host_rate
E	src_bytes	P	num_root	AA	rerror_rate	AL	dst_host_serror_rate
F	dst_bytes	Q	Num of file	AB	srv_rerror_rate	AM	dst_host_srv_serror_rate
G	land	R	Number of shell	AC	same_srv_rate	AN	dst_host_rerror_rate
H	wrong_fragment	S	num_access_files	AD	diff_srv_rate	AO	dst_host_srv_rerror_rate
I	urgent	T	num_outbound_cmds	AE	srv_diff_host_rate		
J	hot	U	Is host login	AF	dst_host_count		
K	num_falied_logins	V	Is guest login	AG	dst_host_srv_count		

Figure 3.1: Feature set of KDD 99 dataset [90]

3.1.2 NSL - KDD

NSL-KDD dataset was created by Tavallaee et al., to address the shortcomings of KDD 99. KDD 99 dataset consists of approximately 78 percent and 75 percent of unessential data records in training and testing sets respectively. [174] note that using the KDD 99 training and testing datasets with machine learning models would lead to a biased result in favor of the majority records that were redundant and hence, minority classes such as U2R attacks which are important for IDS model training would be ignored. They also noted that these redundancies would impact the testing performance as it would be biased towards methods that provide better detection rates for the more frequent data. NSL -KDD is, however, still not an accurate representation of the networks of current times. However, in the absence of good IDS datasets, NSL-KDD is still being widely

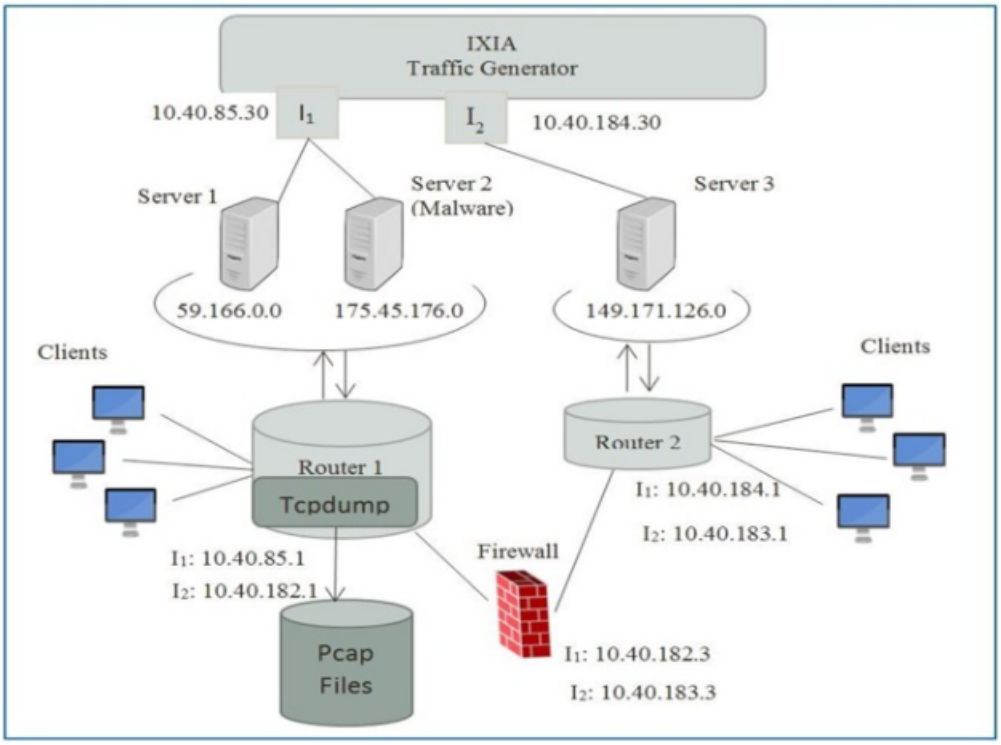

Figure 3.3: Test bed set up for UNSW-NB15 [115]

used by the intrusion detection research community to validate IDS performances. The dataset record distribution for NSL-KDD dataset is shown in Figure 3.2.

3.1.3 UNSW - NB15

UNSW-NB15 is an intrusion detection dataset developed by Dr. Moustafa at the Australian Centre for Cyber Security (ACCS) which is part of the University of New South Wales in Canberra [115]. The main motivation behind generation of this dataset was to address the issues suffered by KDD 99 and NSL-KDD datasets. UNSW-NB15 was created by generating benign and malicious network traffic using the tool called IXIA PerfectStorm and consists of nine attack categories which are listed in Figure 3.5. The test bed set up used for generating UNSW-NB15 dataset is shown in Figure 3.3

#	Name	T	Description
6	*state*	N	The state and its dependent protocol, e.g. ACC, CLO, else (-)
7	*dur*	F	Record total duration
8	*sbytes*	I	Source to destination bytes
9	*dbytes*	I	Destination to source bytes
10	*sttl*	I	Source to destination time to live
11	*dttl*	I	Destination to source time to live
12	*sloss*	I	Source packets retransmitted or dropped
13	*dloss*	I	Destination packets retransmitted or dropped
14	*service*	N	http, ftp, ssh, dns ...,else (-)
15	*sload*	F	Source bits per second
16	*dload*	F	Destination bits per second
17	*spkts*	I	Source to destination packet count
18	*dpkts*	I	Destination to source packet count

Figure 3.4: UNSW-NB15 Basic Features [117]

As listed in Figure 3.5, this dataset includes several modern security vulnerabilities such as fuzzers, backdoors, exploits, reconnaissance, shell code and worms. There are 49 features that are used to describe every record in the datasets. Bro-IDS and Argus tools were used to extract these features from the pcap files generated by the same tools [116]. SQL Server 2008 database was used to log these files and the flow features listed in Figure 3.8 were used to match the features extracted from Bro-IDS and Argus [117]. As seen with NSL-KDD dataset, each of the features were classified into a further four groups based on their types and characteristics as basic features, content features and time features. There is an additional set of features which was grouped under the generic category [74]. These categories of UNSW-NB15 features are shown in Figure 3.4, Figure 3.6, Figure 3.7 and Figure 3.9.

Type	No. Records	Description
Normal	2,218,761	Natural transaction data.
Fuzzers	24,246	Attempting to cause a program or n-etwork suspended by feeding it the randomly generated data.
Analysis	2,677	It contains different attacks of port scan, spam and html files penetrations.
Backdoors	2,329	A technique in which a system security mechanism is bypassed stealthily to access a computer or its data.
DoS	16,353	A malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet.
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.
Generic	215,481	A technique works against all block-ciphers (with a given block and key size), without consideration about the structure of the block-cipher.
Reconnaissa-nce	13,987	Contains all Strikes that can simulate attacks that gather information.
Shellcode	1,511	A small piece of code used as the payload in the exploitation of software vulnerability.
Worms	174	Attacker replicates itself in order to spread to other computers. Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.

Figure 3.5: Data set record distribution of UNSW-NB15 [117]

#	Name	T	Description
19	*swin*	I	Source TCP window advertisement
20	*dwin*	I	Destination TCP window advertisement
21	*stcpb*	I	Source TCP sequence number
22	*dtcpb*	I	Destination TCP sequence number
23	*smeansz*	I	Mean of the flow packet size transmitted by the src
24	*dmeansz*	I	Mean of the flow packet size transmitted by the dst
25	*trans_depth*	I	the depth into the connection of http request/response transaction
26	*res_bdy_len*	I	The content size of the data transferred from the server's http service.

Figure 3.6: UNSW-NB15 Content Features [117]

#	Name	T	Description
27	*sjit*	F	Source jitter (mSec)
28	*djit*	F	Destination jitter (mSec)
29	*stime*	T	record start time
30	*ltime*	T	record last time
31	*sintpkt*	F	Source inter-packet arrival time (mSec)
32	*dintpkt*	F	Destination inter-packet arrival time (mSec)
33	*tcprtt*	F	The sum of 'synack' and 'ackdat' of the TCP.
34	*synack*	F	The time between the SYN and the SYN_ACK packets of the TCP.
35	*ackdat*	F	The time between the SYN_ACK and the ACK packets of the TCP.

Figure 3.7: UNSW-NB15 Time Features [117]

#	Name	T.	Description
1	*srcip*	N	Source IP address
2	*sport*	I	Source port number
3	*dstip*	N	Destination IP address
4	*dsport*	I	Destination port number
5	*proto*	N	Transaction protocol

Figure 3.8: UNSW-NB15 Flow Features [117]

3.1.4 CICIDS - 2017

Gharib et al., [69] in their research, proposed eleven primary characteristics that are pertinent for an IDS dataset framework testing. These include attack diversity, anonymity, available protocols, complete capture, complete interaction, complete network configuration, complete traffic, feature set, heterogeneity, labelling and metadata. As a further extension to the mentioned research, Sharafaldin et al., proposed the Intrusion Detection Evaluation dataset namely CICIDS - 2017 [153] that claims to satisfies all the eleven IDS characteristics listed in [69]. In addition to this, CICIDS - 2017 is a fully labelled dataset with 80 features extracted from the network traffic that were drawn out using the CICFLowMeter [94].

Figure 3.10 explains the test bed framework used for generation of CICIDS - 2017 dataset. In order to generate this dataset, two isolated networks namely Victim-Network and Attack-Network were deployed and the topology of the Victim-Network was designed to reflect the networks of current times. This was achieved by adjusting the

#	Name	T	Description
General purpose features			
36	*is_sm_ips_ports*	B	If source (1) equals to destination (3)IP addresses and port numbers (2)(4) are equal, this variable takes value 1 else 0
37	*ct_state_ttl*	I	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	*ct_flw_http_mthd*	I	No. of flows that has methods such as Get and Post in http service.
39	*is_ftp_login*	B	If the ftp session is accessed by user and password then 1 else 0.
40	*ct_ftp_cmd*	I	No of flows that has a command in ftp session.
Connection features			
41	*ct_srv_src*	I	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	*ct_srv_dst*	I	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	*ct_dst_ltm*	I	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	*ct_src_ ltm*	I	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	*ct_src_dport_ltm*	I	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	*ct_dst_sport_ltm*	I	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	*ct_dst_src_ltm*	I	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).

Figure 3.9: UNSW-NB15 Generic and Connection Features [117]

Victim-Network to include essential and most commonly used devices. B-profile system [151] was used to generate the background traffic, trying to ensure that the background traffic generation would replicate real time traffic. The B profile system considered the hypothetical network behaviour on the basis of various protocols such as HTTP, HTTPS, FTP, SSh and email being used by twenty five users. This dataset has sophisticated

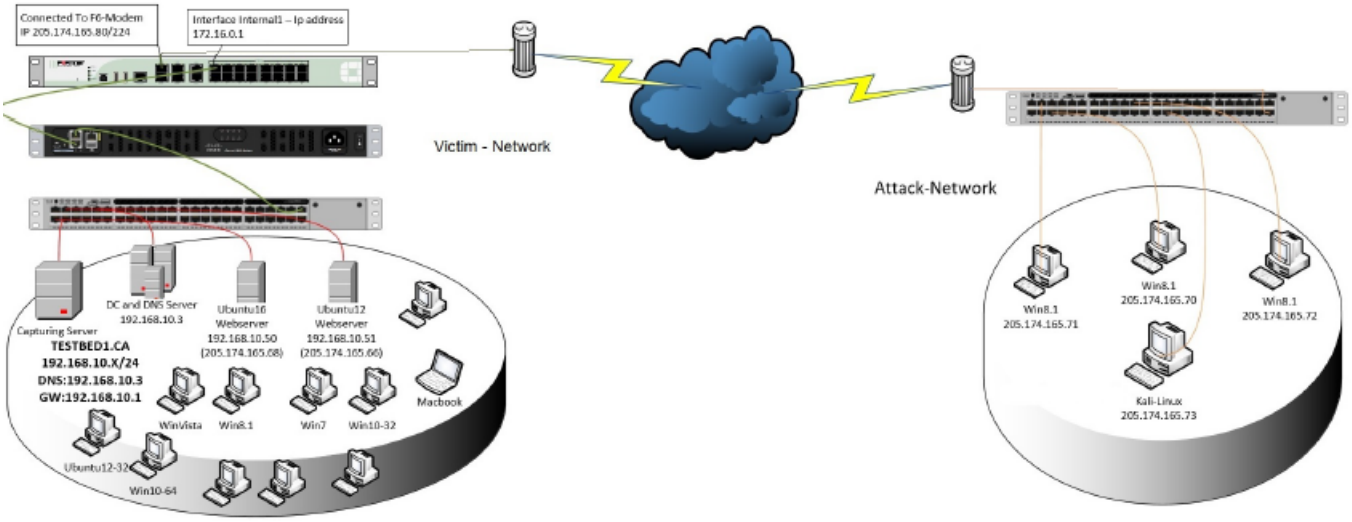

Figure 3.10: Test bed framework for CICIDS - 2017 dataset [153]

and novel attack scenarios like Brute Force attack, Heartbleed attack , Web attack and Infiltration attack along with the popular Dos, DDoS, Botnet attacks. The network data was captured continuously over the period of five days starting on Monday July 3rd until Friday July 7th. During this duration benign traffic was captured and the above mentioned attacks were implemented. The CICIDS - 2017 dataset is publicly available to the research community in pcap format and has labelled flows in the form of eight CSV files for machine learning scenarios [153]. 3.11 gives the details of these csv files, the type of traffic captured and also the number of records in each file.

File Name	Type of Traffic	Number of Record
Monday-WorkingHours.pcap_ISCX.csv	Benign	529,918
Tuesday-WorkingHours.pcap_ISCX.csv	Benign	432,074
	SSH-Patator	5,897
	FTP-Patator	7,938
Wednesday-WorkingHours.pcap_ISCX.csv	Benign	440,031
	DoS Hulk	231,073
	DoS GoldenEye	10,293
	DoS Slowloris	5,796
	DoS Slowhttptest	5,499
	Heartbleed	11
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Benign	168,186
	Web Attack-Brute Force	1,507
	Web Attack-Sql Injection	21
	Web Attack-XSS	652
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Benign	288,566
	Infiltration	36
Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign	189,067
	Bot	1,966
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Benign	127,537
	Portscan	158,930
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Benign	97,718
	DdoS	128,027
Total Instance/ Record		**2,830,743**

Figure 3.11: Data set record distribution of CICIDS - 2017 [163]

Figure 3.12 describes some of the attack categories that are part of the CICIDS- 2017 dataset.

Class	Description
Normal	Normal connection records
SSH-Patator	Secure shell - Representation of brute force attack
FTP-Patator	File transfer protocol - Representation of brute force attack
DoS	Intruder aims at making network resources down and consequently, resources are inaccessible to authorized users
Web	Attacks are related to web
Bot	Hosts are controlled by bot owners to perform various tasks such as steal data, send spam and others
DDoS	Distributed Denial of Service ('DDoS') is an attempt made to make services down using multiple sources. These are achieved using botnet
PortScan	Port scan is used to find the specific port which is open for a particular service. Using this attacker can get information related to sender and receiver's listening information

Figure 3.12: CICIDS 2017 Dataset Attack Category [188]

The list of features available in CICIDS 2017 dataset is shown in Figure 3.13. We further describe the dataset record details for CICIDS -2017, UNSW - NB15 and NSL - KDD datasets used for our experiments in Chapter 5.

No.	Feature	No.	Feature	No.	Feature
1	Source Port	28	Bwd IAT Total	55	Average Packet Size
2	Destination Port	29	Bwd IAT Mean	56	Avg Fwd Segment Size
3	Protocol	30	Bwd IAT Std	57	Avg Bwd Segment Size
4	Flow Duration	31	Bwd IAT Max	58	Fwd Avg Bytes/Bulk
5	Total Fwd Packets	32	Bwd IAT Min	59	Fwd Avg Packets/Bulk
6	Total Backward Packets	33	Fwd PSH Flags	60	Fwd Avg Bulk Rate
7	Total Length of Fwd Pck	34	Bwd PSH Flags	61	Bwd Avg Bytes/Bulk
8	Total Length of Bwd Pck	35	Fwd URG Flags	62	Bwd Avg Packets/Bulk
9	Fwd Packet Length Max	36	Bwd URG Flags	63	Bwd Avg Bulk Rate
10	Fwd Packet Length Min	37	Fwd Header Length	64	Subflow Fwd Packets
11	Fwd Pck Length Mean	38	Bwd Header Length	65	Subflow Fwd Bytes
12	Fwd Packet Length Std	39	Fwd Packets/s	66	Subflow Bwd Packets
13	Bwd Packet Length Max	40	Bwd Packets/s	67	Subflow Bwd Bytes
14	Bwd Packet Length Min	41	Min Packet Length	68	Init_Win_bytes_fwd
15	Bwd Packet Length(avg)	42	Max Packet Length	69	act_data_pkt_fwd
16	Bwd Packet Length Std	43	Packet Length Mean	70	min_seg_size_fwd
17	Flow Bytes/s	44	Packet Length Std	71	Active Mean
18	Flow Packets/s	45	Packet Len. Variance	72	Active Std
19	Flow IAT Mean	46	FIN Flag Count	73	Active Max
20	Flow IAT Std	47	SYN Flag Count	74	Active Min
21	Flow IAT Max	48	RST Flag Count	75	Idle Mean
22	Flow IAT Min	49	PSH Flag Count	76	Idle packet
23	Fwd IAT Total	50	ACK Flag Count	77	Idle Std
24	Fwd IAT Mean	51	URG Flag Count	78	Idle Max
25	Fwd IAT Std	52	CWE Flag Count	79	Idle Min
26	Fwd IAT Max	53	ECE Flag Count	80	Label
27	Fwd IAT Min	54	Down/Up Ratio		

Figure 3.13: CICIDS 2017 Dataset Features [20]

3.2 Feature Selection using Machine Learning techniques

Feature selection is an important step in which machine learning classifiers are employed for the intrusion detection process and pattern recognition. The aim of the feature selection step is to identify the most suitable dataset features and to remove the remaining unnecessary and redundant features from the dataset as these noisy features

can introduce bias and inadequate overview of the intrusion classification model [38] [142]. Feature selection process is a significant step for high dimensionality datasets as it reduces the size of the dataset, which further enhances the machine learning classifiers to train effectively and speedily [86]. Thus the anomaly classification can be achieved from a relatively smaller number of features which consist of the majority of the relevant information corresponding to each class [38]. Feature selection algorithms are classified into three groups namely filter, wrapper and hybrid. Hybrid feature selection methods are also known as embedded techniques for feature selection[8].

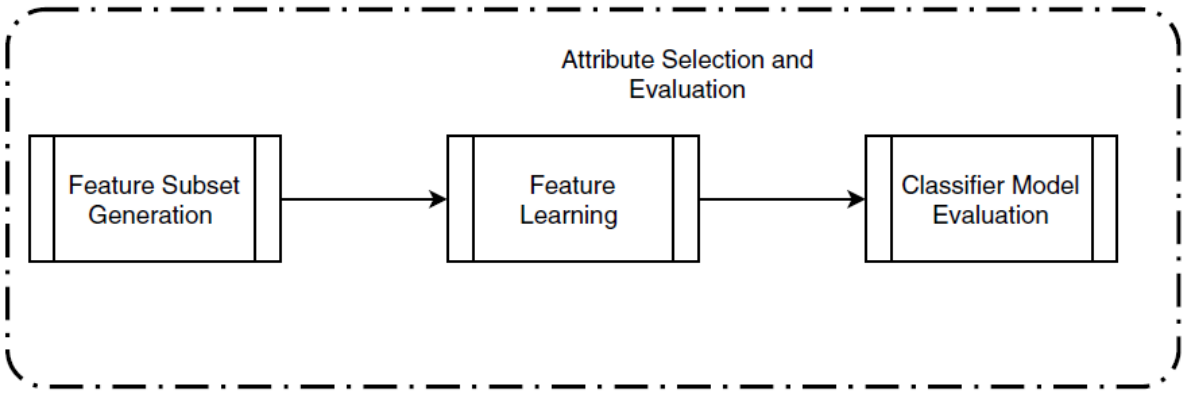

Figure 3.14: Wrapper based Feature Selection Method [177]

Wrapper methods produce an optimal feature subspace by employing searching methods. Further, classification algorithms are used to assess the relevance of the selected feature subset using classification rate as the measuring criterion. Wrapper based methods are dependent on the learning algorithm as they use threshold for classification rate. The feature subset with best classification rate value is selected [111]. However, these methods suffer from the major drawback of over fitting as the learning classifier is trained several times [26]. In addition, due to the iterative feature subset collection, wrapper based methods have large computational time [83]. Figure 3.14 illustrates the wrapper-based feature selection method [177]. Considerable contemporary research work using wrapper-based feature selection approach has been done up until now for intrusion detection systems. It is noteworthy that the majority of the wrapper-based feature selection frameworks make use of meta heuristic and nature inspired machine learning methods like Genetic Algorithm (GA) [88],[187], Cuttlefish algorithm (CFA) [59], Particle Swarm Optimization (PSO) [19], [109] and Multi-Objective Genetic Algorithm (MOGA) [49] for the searching process and create a low dimension IDS dataset.

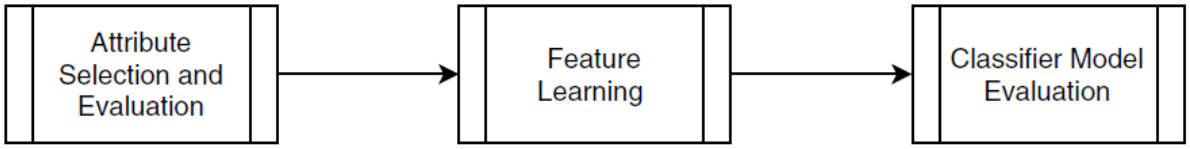

Figure 3.15: Filter based Feature Selection Method [177]

Filter based methods, on the other hand, generate optimal feature subset using the intrinsic properties of the dataset instances and do not depend on the choice of the learning algorithm. These methods are speedy and are robust to counter the issue of over fitting that is encountered by the wrapper methods. However, the drawback with these methods is that, the performance of the learning algorithm is not balanced with respect to the generated feature subset [111]. Figure 3.15 illustrates the filter based feature selection method [177]. Some of the research work done using filter based feature selection includes [7] , [28], , [81], [120] and [191]. The first two studies focus on using mutual information (MI) to select the optimal features. [24] is one of the primary works that suggested using MI for filter based feature selection and many researchers like [93] and [10] have tried to improvise on the feature selection algorithm introduced by Battiti [24].

Hybrid feature selection methods are also called embedded methods. They combine the concept of wrapper-based methods and filter-based methods. Thus the hybrid feature selection approach integrates the benefits of wrapper and filter based approaches, resulting in improvised detection performance. This is achieved by applying the attribute selection process during the training phase. Thus, the classifier training iterations for each attribute subset is reduced by the optimum utilization of data [111]. Figure 3.16 illustrates the hybrid feature selection method [177]. Ambusaidi et al. [8] developed a hybrid FS approach Improved Forward Floating Selection (IFFS), which involves two stages. The first stage uses the filter method with mutual information (MI) for sorting and removing features based on the rank and the second stage uses wrapper-based feature selection to further generate the optimal feature set. The research study by [119] proposes another hybrid feature selection method which uses Central points (CP) method and Association Rule Mining (ARM) method to develop an intrusion detection system with low False Alarm Rate (FAR). [133] came up with a new hybrid feature selection approach based on combining Correlation Feature Selection (CFS) and Support Vector

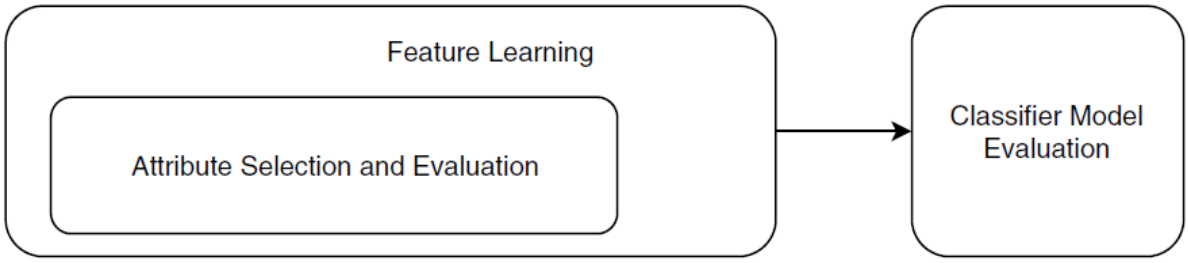

Figure 3.16: Embedded Feature Selection Method[177]

Machines (SVM) while [160] propose to blend of chi-square algorithm with Random Forest (RF) method to build a hybrid feature selection scheme. As discussed in [26], these methods involve wrapper method as the second stage of feature selection which uses the results obtained through filter based feature selection method in the first stage. Hence, the major drawback of hybrid FS schemes is that the wrapper method may not be able to monitor some relevant features that have been filtered in stage one. Zhou et al., [193] propose a novel hybrid FS method called CFS-BA which uses meta heuristic Bat Algorithm (BA) and correlation feature selection. CFS - BA selects the best feature set from the several sets generated by the BA algorithm and further correlation filter is employed to evaluate the selected optimal solution. The best solution is updated with every iterations until the repetitive process ends. Many authors such as [12] and [38] have provided comprehensive comparative study on the above mentioned feature selection methods. Figure 3.17 enlists the advantages and disadvantages of Filter-based, Wrapper-based and Hybrid feature selection methods.

In spite of much research done in the feature selection space, there is no single feature selection approach that can be considered better than the rest. This has motivated several researchers to investigate ensemble-based feature selection methods for intrusion detection. Even though machine learning research has plenty of studies such as [50], [80], [80], [82], [129], [179] and [180] that employ ensemble-based feature selection techniques for various data mining applications, not much work has been focused on using the ensemble feature selection method for intrusion detection and anomaly classification problems. Ensemble-based feature selection methods integrate the outputs of a number of FS methods and hence, provide a more powerful feature selection approach which benefits intrusion detection schemes using high dimensional IDS datasets for performance evaluation. The hypothesis behind using more than one feature selection algorithm in an ensemble, is that different feature selection methods will generate optimal subsets

Feature selection method	Advantages	Disadvantages
Filter based	Interaction with the classifier	It is not dependent on the classifier
	Computationally cost effective	
	Good generalization ability	
Wrapper based	Interaction with the classifier	It is dependent on the classifier used.
	Computationally cost effective	
	It derives the feature dependencies	
Embedded	Interacts with the classifier algorithm	It is costly in terms of computation
	It derives the feature dependencies	It is dependent on the classifier used

Figure 3.17: Advantages and Disadvantages of Feature Selection Methods[177]

and by combining the output of these methods using ranking combination method (RCM) or Subset Combination method (SCM) [26], a highly efficient feature subset is derived, which is robust for various IDS datasets [77].

Chebrolu et al., [39] proposed an ensemble-based feature selection scheme with Bayesian network classifier and CART algorithm which makes use of the contradictions between the incorrectly classified results to refine the performance of the IDS model. Similar works such as [106], [130] and [150], combine multiple methods like Information Gain (IG), Correlation filter, ReliefF filter, chi-square, gain ratio, decision tree and Naive Bayes algorithm, for feature selection and have reported optimistic results for intrusion detection schemes. Binbusayyia et al., [26] have proposed an ensemble feature selection approach which merges the outputs of ReliefF Filter, Information Gain, Consistency based Feature Selection (CBF) and Correlation based Feature Selection (CFS).

3.3 Proposed Ensemble-based Feature Selection Approach using Machine learning methods

In recent times, researchers are motivated to investigate ensemble-based feature selection methods with the aim to address the challenge of choosing the best feature selection approach. This is due to the advantage that ensemble feature selection provides with its ability to combine the outputs of multiple FS methods, thus ensuring that the feature subset generated contains the most informative features from the original IDS dataset.

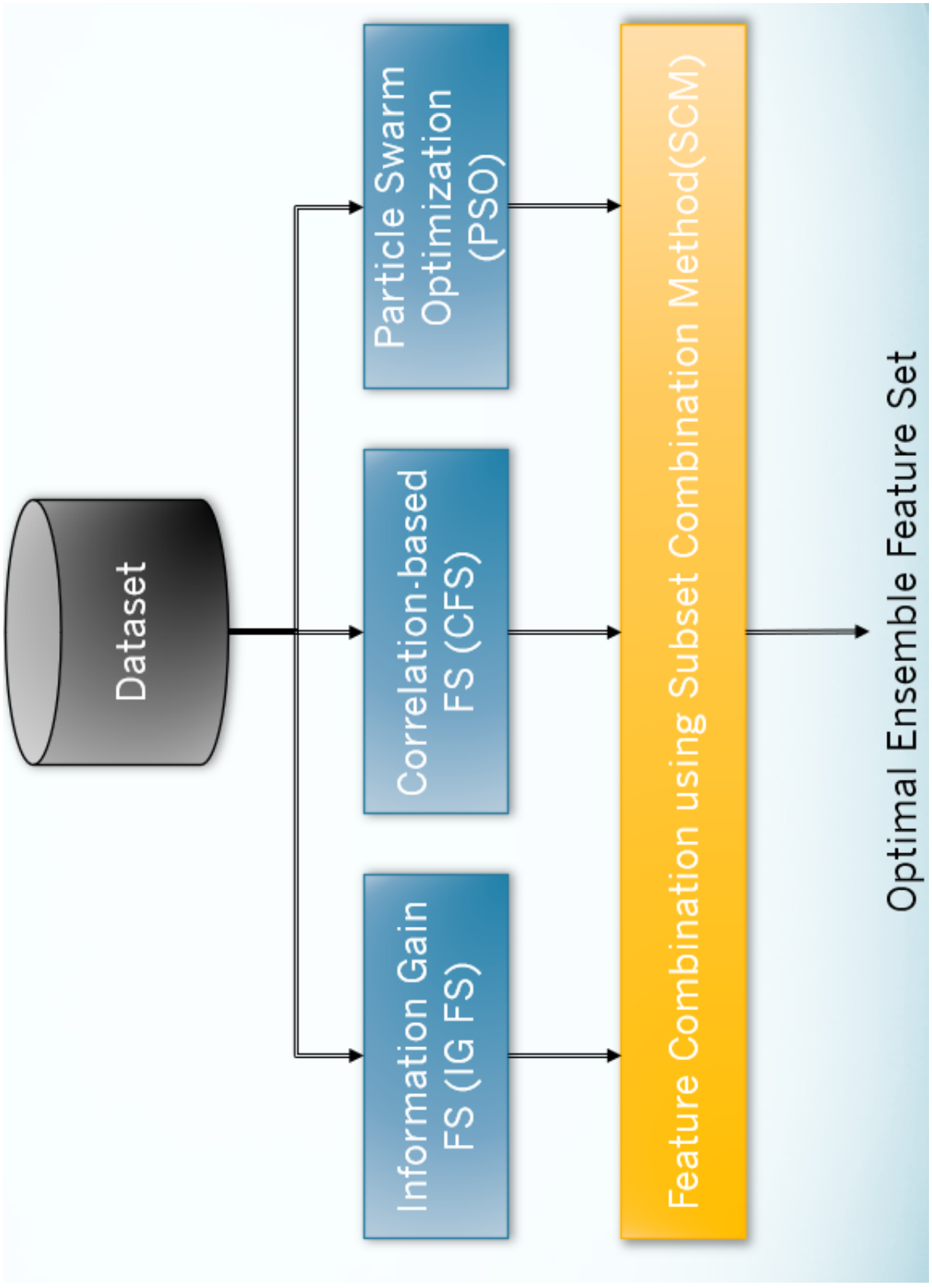

Figure 3.18: Proposed Ensemble-based Feature Selection approach

This is the first challenge as noted in Section 1.2 of Chapter 1. We address this in our thesis by proposing the novel framework for our ensemble based feature selection approach. This scheme is shown in Figure 3.18.

The proposed feature selection method uses three feature selection methods namely (1) Information Gain based feature selection algorithm (2) Correlation Based Feature Selection algorithm and (3) Particle Swarm Optimisation (PSO) technique to generate finest feature subset. We provide a concise theoretical description for each of the three feature selection algorithms used in our proposed FS method below:

1. Information Gain Based Feature Selection (IG): This feature selection method searches for the optimum features by calculating the entropy for each feature and ranking all the features using this entropy value. Information Gain method is a filter based feature selection technique and also comes under the uni variate method category. Feature entropy is calculated using the below formula :

$$Entropy(S) = \sum_{i}^{c} -P_i log_2 P_i \tag{3.1}$$

where c equals the total number of classification values and P_{i} denotes the total number of instances for the class i. The above feature entropy value is used to further calculate the Information Gain using the following formula [163] :

$$Gain(S,A) = Entropy(S) - \sum_{Values(A)} \frac{|Sv|}{|S|} Entropy(S_v) \tag{3.2}$$

where S is the number of features, A is the attribute, v is the value for attribute A. *Values(A)* represents the set of all possible values for attribute A, $|Sv|$ is the number of samples for value v and the number of features for all data instances is $|S|$. *Entropy (Sv)* is the entropy for features which have the value v [163].

2. Correlation Based Feature Selection (CFS): This feature selection aims to select the the features that are highly correlated with the target class but have no correlation with each other [193]. The merit for a feature subset that has the number

of features equal to k, is calculated using the Pearson's correlation which has the formula as below [41] :

$$Merit_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(r-1)\bar{r}_{ff}}} \tag{3.3}$$

where $\bar{r}_{cf}$ equals the average feature class correlation and $\bar{r}_{ff}$ is the average feature-feature correlation [41].

3. Particle Swarm Optimization (PSO): Research related to meta heuristic, population-based optimization methods for feature selection problems, has become prominent within the research community in recent times. These methods fall under the umbrella of evolutionary computation (EC). One of the widely used approaches from the EC family is Swarm Intelligence (SI) which is comprised of several algorithms which are motivated by behavioural patterns observed in animals and insects while interacting socially in nature. SI algorithms are gaining popularity for feature selection tasks as they are easy to implement and are efficient search methods inspired by nature for generating best feature subsets from high dimensional data [126]. Ant Colony Optimization (ACO), Artificial Bee Colony Optimization (ABC) [29] and Particle Swarm Optimization (PSO) [70] are prominent algorithms from the SI family. Particle Swarm Optimization (PSO) was developed in 1995 by researchers Eberhart and Kennedy and mimics the social communication behaviours seen in creatures like birds and fish, that move as a swarm in nature [58]. Implementation of PSO based feature selection needs to be actively investigated in solving the dimensionality curse for IDS datasets as this approach has several benefits like effortless implementation, fast search convergence rate to obtain an optimum solution, performance consistency, comparatively low memory usage and stability [43].

   In the PSO method, the initial swarm (*population*) of particles (*candidate solutions*) is generated randomly. Further, each particle notes its velocity and location which is considered its personal best solution (*pbest*) in the given feature space. The best solution found by the swarm (*gbest*) is also recorded. *pbest* and *gbest* are combined

to obtain information that is used to guide the swarm further in the given search space. PSO feature selection is an iterative process and, with each iteration, the velocity is updated using the following equation [18] :

$$v^i[t+1] = w.v^i[t] + c_1 r_1(p^{i,best}[t] - p^i[t]) + c_2 r_2(p^{g,best}[t] - p^i[t]) \quad (3.4)$$

Position of the particle is updated using the following equation [18] :

$$p^i[t+1] = p^i[t]) + v^i[t+1]) \quad (3.5)$$

where, i = 1,2, ..., N.
N = Swarm population number.
v^{i}[t] = Velocity vector in [t]$^{\text{th}}$ iteration.
p^{i}[t] = Current location of the t^{th} particle.
$p^{\text{i,best}}$[t] = Previous best location of the t^{th} particle.
$p^{\text{g,best}}$[t] = Previous best location of the entire swarm.
w = Parameter managing local and global search pressure.
c_1 *and* c_2 = Acceleration coefficients.
c_1 = Cognitive parameter and c_2 = Social parameter.
r_1 *and* r_2 = Random number between [0,1] [18].

Each of the methods, Information Gain, Correlation based and Particle Swarm Optimization algorithm; generates individual feature subsets for the CICIDS - 2017. These three individual feature subsets are further combined using the Subset Combination Method (SCM) [26] which chooses the features that are available in a minimum of two of the three individually generated feature subsets that were produced by applying IG, CFS and PSO algorithms separately. The proposed ensemble-based feature selection technique, hence, combines the most important features selected by each of the individual feature selection methods and generates a ensemble feature subset that represents only those features from the original CICDS - 2017 dataset that provide the most information. Thus, the proposed ensemble-based feature selection technique reduces the high dimensional CICIDS - 2017 dataset effectively and reduces the noise which otherwise would have had adverse impact on the prediction performance of anomaly detection. We further reduce the learning time for our ensemble classifier model in addition to other resource overheads [26]. We explain the generated ensemble feature set for CICIDS - 2017 and NSL - KDD datasets in Chapter 5, in addition to a comparative analysis of performance

of our proposed IDS scheme (explained in Chapter 4) without feature selection and with ensemble-based feature subset generated using our proposed FS approach.

3.4 Chapter Summary

In this chapter we have described publicly available old intrusion detection datasets namely KDD 99 and NSL - KDD. We have highlighted the issues that these datasets have when used for validation of modern intrusion detection schemes and we have further argued the need for a more recent intrusion detection dataset that mirrors the modern network systems and includes the latest network threats. We have identified the latest intrusion detection dataset namely CICIDS - 2017 which satisfies the requirement for diverse and modern day attack classes that are not addressed by KDD 99 and NSL-KDD. The features of the dataset has been elaborated and the attacks covered by these datasets have also been mentioned. Commonly used performance metrics for IDS evaluation have also been described in this chapter. We have further discussed the significance of feature selection methods with machine learning algorithms used for intrusion detection schemes. Ensemble based feature selection approach has been explained in this chapter. In addition, we have presented our proposed ensemble based feature selection approach that employs three feature selection algorithms namely Information Gain (IG), Correlation based Feature Selection (CFS) and Particle Swarm Optimization (PSO) and have given a short theoretical background for these methods. We have explained the commonly used performance evaluation metrics for intrusion detection models and have identified the ones that will be used for performance evaluation of our proposed ensemble based IDS scheme. Hence, we have successfully addressed the first problem statement mentioned in Section 1.2 of Chapter 1, which stated that there was a gap in feature selection techniques due to unavailability of ensemble based feature selection methods for IDS dimensionality reduction in literature.

CHAPTER 4

PROPOSED ENSEMBLE BASED MACHINE LEARNING TECHNIQUE FOR INTRUSION DETECTION MODEL

Machine learning (ML) is a sub-domain of Artificial Intelligence (AI). Machine learning techniques are widely used in areas such as image processing, medical sciences, finance, information technology and cyber intrusion detection. ML approaches are widely applied to building IDS models for optimizing anomaly network intrusion detection. This is due to the ability of ML algorithms to learn the specific behaviour of the network from the training data and progressively adapt when the data instances increase. These methods are used extensively for intrusion classification and prediction problems [35]. ML algorithms can be further classified into three categories: *(1) Supervised ML, (2) Unsupervised ML* and *(3) Semi-supervised ML* methods [149].

Supervised ML algorithms are used for datasets with labelled classes while *unsupervised ML algorithms* are used for datasets for which the data class has no labels. Supervised ML algorithms need to be trained on using input dataset features and further the trained model is used to classify or predict on testing data with predefined output classes. The performance of supervised ML models is measured by the accuracy or the prediction or classification of the output class. When an admissible value of model performance is achieved, the learning process for the ML algorithm comes to an end [6]. Decision tree (DT), Naive Bayes (NB), multi-class Support Vector Machines (SVM), Multilayer Percetptron (MLP), Back Propagation Artificial Neural Network (BP-ANN)

and K Nearest Neighbor (KNN) are some of the popular supervised ML techniques used in research for anomaly network intrusion detection [111] [149].

Unsupervised ML algorithms, in contrast, work with unlabelled data for recognizing patterns within the entire dataset for clustering and deriving correlation within the dataset instances to identify the relationships within the dataset. Hence, these algorithms can be used for generating labels for unlabeled datasets using clustering methods. Principal Component Analysis (PCA), Self Organizing Map (SOM), DBSCAN and k-means clustering are the commonly used unsupervised ML methods [6] [111]. *Semi-supervised ML algorithms* use both supervised and unsupervised ML methods for generating the detection model using data that has unlabelled data instances as well as some labeled data records [149].

In section 4.1, we provide an overview of the various ensemble-based intrusion detection techniques used in current research. Section 4.2 explains the proposed ensemble-based network anomaly intrusion detection framework and briefly explains the machine learning algorithms used in our scheme. Section 4.3 gives the chapter summary.

4.1 Ensemble-based Machine Learning approach for Network Anomaly Detection - An Overview

With the advancement in IDS research using ML approaches for anomaly network IDS, it is observed that choosing a single ML classifier to build a robust IDS with efficient detection performance may not be the best approach [193]. In order to address this weakness with single ML classifier based NIDS, researchers have enthusiastically used Ensemble-based ML approach. Ensemble-based method is a supervised machine learning algorithm. It employs more than one ML algorithm, which are called base learners or base classifiers. Further, the outcome from each of these base learners is combined to provide the intrusion classification. Thus, individual ML classifiers are used to first create a set of presumptions which are combined by the ensemble classifier to find the best possible solution. Ensemble approach is a supervised ML algorithm [111] [194].

4.1.1 Techniques to combine ensemble classifiers

Ensemble approaches can be further classified as homogeneous ensemble classifier and heterogeneous ensemble classifier [91]. Homogeneous ensemble methods use the same base learner multiple times over multiple, non-identical subsets of the same training data while heterogeneous ensemble methods use different machine learning classifiers to train on a single training dataset and their individual prediction outcomes are then combined to generate the unique anomaly classification. The techniques used to combine the base classifiers in the ensemble approach are (1) Bagging (2) Boosting (3) Stacking and (4) Voting. Bagging and boosting techniques are used with homogeneous ensemble methods, while stacking and voting techniques are used with heterogeneous ensemble methods [3].

Bagging is also known as bootstrap aggregation. It is used with ML methods which tend to have a greater variance which can result in over fitting of the prediction model. Bagging is used to reduce the variance by using a model-averaging method [35]. Using the bagging method, different subsets of the dataset are created and the ensemble classifier is trained on each of these sample training datasets. The outcome for each of the subset modelling is further combined using voting or averaging technique [33] [111].

Boosting is used to boost the performance of a weak ML classifier whose prediction is not any better than making a random guess [23]. The weak classifier is trained using random data subsets from the training datasets and each subset has no overlapping records between them for the first iteration. In the second iterations, the data subset includes new training samples and 50 percent of those instances that were incorrectly classified in the first iteration. At the end of the iterative process, majority voting method is used to generate the final outcome of the predictive model [111].

Stacking is used with multiple base learning classifiers and involves two stages namely base learning and meta-learning. In the first step, a new training dataset is generated after the initial base learner is trained on the original trained dataset and this is further used to train the meta learner classifier. The anomaly prediction of the test dataset is done using the previously trained meta-classifier [139].

Majority voting is also a meta learning algorithm and is used with multiple base learning classifiers which is similar to stacking. However the outcome of the base learner classifiers is obtained by using a combination rule such as majority voting, minimum probability, maximum probability, product of probabilities and average of probabilities. [193].

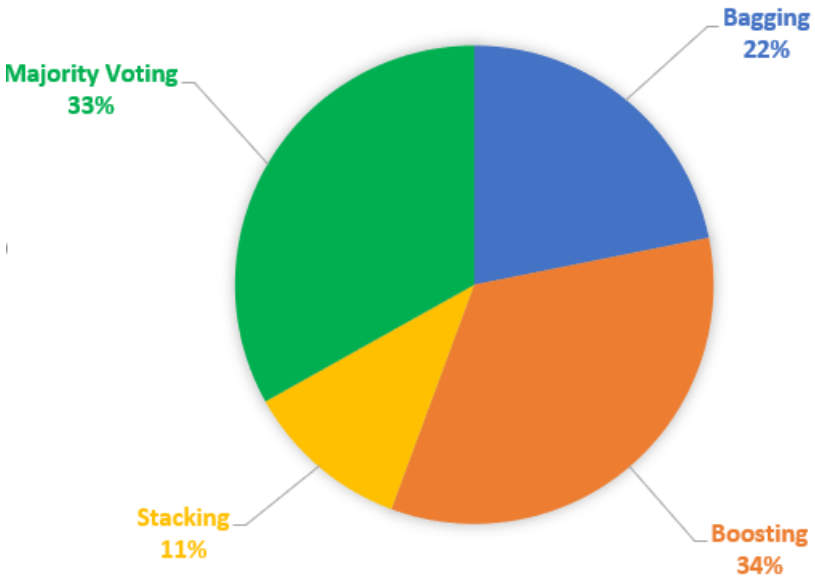

Figure 4.1: Distribution of Ensemble Base Classifier combining method [170]

Tama et al., [170] have done a mapping study for ensemble-based learning methods using 124 research papers from the literature available on ensemble IDS. We used the information provided in their paper to visualize the distribution of ensemble methods for combining base classifiers and the combination rules for majority voting which is shown in Figure 4.1 and Figure 4.2 respectively. As seen in Figure 4.1, boosting is the the most commonly used method for combining base learners followed by majority voting technique. However, as mentioned earlier, boosting is used with homogeneous family of ensemble and majority voting works with heterogeneous ensemble methods. In Figure 4.2, which illustrates the various combining methods for base ensemble classifiers, it is noted that majority voting technique is used extensively.

In [40], Chebrolu et al., used Bayesian networks (BN) and Classification and Regression Trees (CART) to develop an ensemble based hybrid technique. Chan et al. [36],

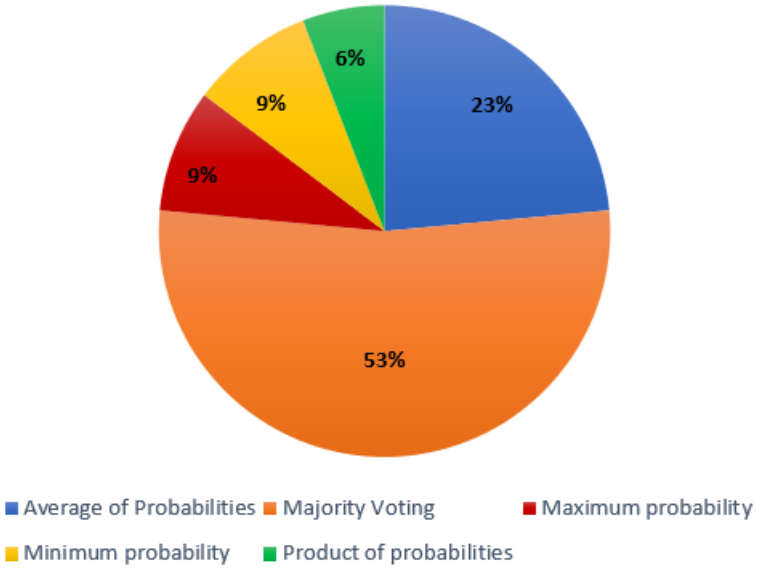

Figure 4.2: Distribution of Combination Rule for Majority Voting using data from [170]

proposed an ensemble-based scheme using Multi-Layer Perceptron (MLP), Radial Basis Function Neural Network (RBF - ANN) and Support Vector Machine (SVM) as base classifiers. Multiple classifier combination schemes, namely majority voting, weighted majority voting, stacking with Naive Byes and ANN, Dempster- Schafer combination and averaging posterior probability, were used to classify normal and malicious instances for KDD 99 dataset. Their work focused on Denial-of-Service (DoS) attacks. Mukkamala et al. [121], employed six ensemble base learners, namely ANN, SVM, ANN(RP), ANN(SCG), ANN(OSS) and MARS; the individual results of each of these classifiers was combined to generate an ensemble classifier and DARPA 98 was used to test the performance. Borji [30] demonstrated that the performance of his proposed ensemble method, which was created using ANN, SVM, C4.5 and KNN as base classifiers and combination rule namely, majority voting, averaging of posterior probabilities and belief measurement, was better than the individual classifiers. A multi class detection ensemble model was presented by [76], where boosting was used with Random Forest to build an ensemble classifier. The model was tested using KDD 99 and the reported results were superior to all ML methods such as Naive Bayes and KNN. Folino et al. [63], used Genetic Programming (GP) to construct a distributed ensemble learning algorithm for network intrusion detection . Nguyen et al. [127] used k-means clustering to create an ensemble model and efficiency was measured by its performance on local data segments which were created using k-means clustering. Bahiri et al. [16], used a novel algorithm called the

Greedy-Boost method to create a hybrid ensemble approach with the help of AdaBoost, C4.5 and Greedy Boost and used KDD 99 for validation of their model. They claimed that their proposed ensemble algorithm, detected better precision values for attacks including probing, U2R and R2L attacks. The study by Malik et al. [105]. used the meta heuristic ML algorithm called Binary Particle Swarm Optimization (BPSO) and Random Forest to generate the ensemble classifier which was used to predict Probe attacks in KDD 99 dataset. [100] proposed using majority voting method to combine rotation forest for SVM based ensemble scheme. A systematic combination of bagging and boosting algorithm was used by [73] to present a hybrid ensemble approach. This is one of the few studies that chose to implement feature selection technique for NSL-KDD dataset. Majority voting method was used to generate classification for the proposed ensemble classifier. Tame et al. [172] presented a binary classification ensemble method using majority voting and averaging of posterior probabilities with base classifier algorithms namely, C4.5, Random Forest and CART. The model performance was validated using NSL-KDD dataset for which they also employed a feature selection method using PSO and Correlation Based FS. Multi-Layer Perceptron (MLP) was employed to create a binary and multi-classification ensemble model for deep learning with neural networks by Vinaykumar et al. in [189]. The authors used KDD - 99 and NSL - KDD datasets for model validation. Bansal and Kaur [20] used XGBOost to compare the performance of their model to detect DoS attacks, with several other ML algorithms like Adaboost, MLP, NB, and HMM. On the basis of their performance results, they claim that XGBoost ensemble classifier is the most efficient as compared to other ML algorithms used in their paper.

Research studies using ensemble models that were published in the last one year have also been mentioned as follows: Nanda et al. [123] propose an ensemble-based framework that uses mean squares error for packet payload aggregation and Bayes algorithm for prediction with the aim to solve the problem of data breach. Rajadurai and Gandhi [139] used stacking method with base classifiers namely, Random Forest and Gradient Boosting Machine (GBM), to generate an ensemble model. They also used several other algorithms such as CART, ANND and XGBoost to compare their model. NSL - KDD dataset was used for model validation. Tama et al. [171], also employed stacking ensemble method for anomaly detection in web applications. Stacking method was also implemented by Abriami et al. [2] in which they blended RF, SVM and NB algorithms in stage one of stacking and in stage two, the meta classifier ensemble IDS was generated using linear regression method. Finally, we conclude this literature review on ensemble

methods with the study by Zhou et al. [193], who used Correlation Feature Selection (CFS) and Bat Algorithm (BA) for generating best features from input datasets and used the new feature subset with their proposed ensemble scheme. They used majority voting method for ensemble algorithm to combine the base classifiers namely, C4.5, Random Forest and Forest PA. Their proposed model was tested for DDoS attack prediction. These research works are comprehensively presented in Table 4.1 which shows the comparison of the ensemble-based network anomaly detection schemes that have been discussed above.

Table 4.1: Comparison of Ensemble-based Network Anomaly IDS schemes mentioned in this work

Begin of Table				
Ensemble Method	Year	FS method	Base Classifiers	Dataset used
Hybrid method (Chebrolu et al. [40])	2005	–	NB, CART	KDD99
Voting, Stacking Dempster-Schafer (Chan et al. [36])	2005	–	MLP, RBF-ANN, SVM	KDD99
Majority Voting (Mukkamal et al. [121])	2005	–	ANN,SVM,MARS	DARPA 98
1 class SVM (Perdisci et al. [136])	2006	–	SVM	Operational Points
Majority Voting (Borji [30])	2007	–	ANN, SVM, C4.5, kNN	DARPA 98
McPad Model (Perdisci et al. [135])	2009	–	1 class SVM	DARPA 98
Boosting (Gudadhe et al. [76])	2010	–	Decision Tree	KDD99
GEdIDS Model (Folino et al. [63])	2010	–	Genetic Programming	KDD 99
Clusterig (Nguyen et al. [127])	2011	–	like this	KDD99
Greedy-Boost (Bahri et al. [16])	2011	–	C4.5	KDD99

Continuation of Table 4.1

Ensemble Method	Year	FS method	Base Classifiers	Dataset used
Majority Voting (Malik et al. [105])	2011	BPSO	Random Forest	KDD99
Majority Voting (Lin et al. [100])	2012	–	SVM	KDD99
Majority voting (Govindarajan et al. [73])	2012	BFS	RBF, SVM	NSL-KDD
Average of Probabilities, Majroity Voting (Tame et al. [172])	2015	PSO, CFS	C4.5, Random Forest, CART	NSL-KDD
Multi-Layer Perceptron (Vinaykumar et al. [189])	2017	–	MLP	Self Generated Dataset
XGBoost (Bansal et al. [20])	2018	–	Adaboost, MLP, NB, MLP	CICIDS-2017
Clustering (Nanda et al. [123])	2020	–	HMM	like this
Stacking (Rajadurai et al. [139])	2020	–	ANN, CART, Random Forest, SVM	NSL-KDD
Stacking (Tama et al. [171])	2020	–	Random Forest, gradient boosting, and XGBoost	NSL-KDD, UNSW-NB15, CICIDS-2017
Least Square Support Vector Machine (Abirami et al. [2])	2020	PCA, Random Forest	Random Forest, SVM, NB	KDD99, NSL-KDD, Kyoto 2006
Majority Voting (Zhou et al. [193])	2020	CFS+BA	C4.5, Random Forest, Forest PA	NSL-KDD, AWID, CICIDS-2017
Majority Voting (Our Propose Model)	–	Ensemble of Information Gain, Correlation, PSO	KNN, C4.5/J48, Random Forest	NSL-KDD, CICIDS-2017

End of Table

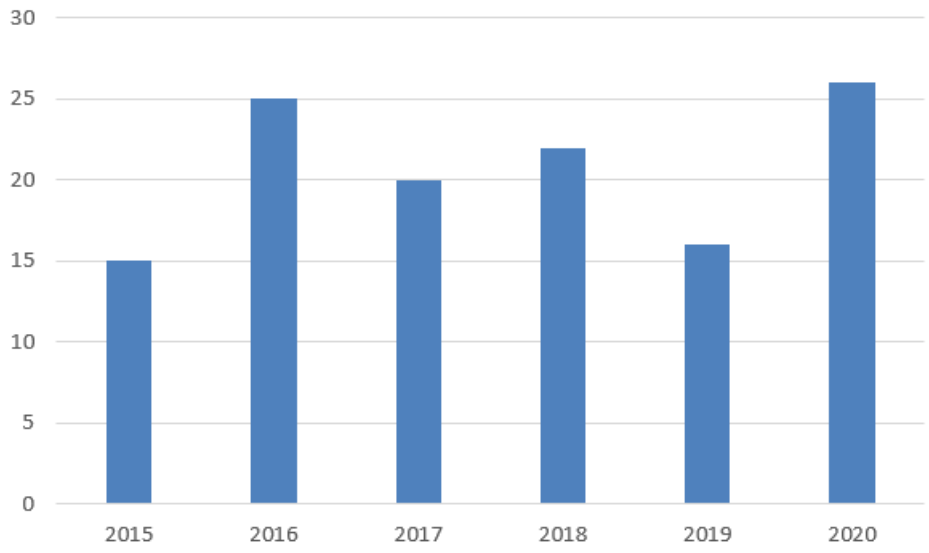

Figure 4.3: Trend of using Ensemble-based methods between 2015 - 2020 [170]

Authors, Tama and Lim have published a list of ensemble-based research papers from the past five years, which covers 124 well known journal, conference papers and workshops. Using these publications we present Figure 4.3 that shows the advancement of ensemble-based intrusion detection schemes from 2015 to 2020 [170]. Figure 4.3 indicates that since 2016, the research employing ensemble-based IDS techniques were not as widely explored until 2020, when once again these methods have gained popularity. The ability of ensemble-based methods to overcome the drawbacks of weak machine learning IDS classifiers and further combine the strengths of multiple machine learning classifiers to yield an optimal and robust IDS classifier model, is making them a popular choice within the IDS research domain.

4.2 Proposed Ensemble-based Network Anomaly Detection Intrusion Detection Framework

In this section we present an ensemble-based intrusion detection scheme for network anomaly detection. Our proposed framework is illustrated in Figure 4.4. Generally, every intrusion detection system has four parts : data source, data pre-processing, decision engine and defense response [116]. Our proposed IDS framework involves data pre - processing, ensemble-based feature selection and ensemble-based IDS scheme, we have

focused on data pre-processing and decision engine components specifically. The data pre-processing module of the framework is shown in Figure 4.4, is explained below.

4.2.1 Data Pre-processing

Data preprocessing is a very important step for machine learning and data mining. Publicly available IDS datasets are generated by capturing real time network traffic from diversified sources. This results in a sizable number of redundant, noisy, incompatible and missing trace records [97]. Redundant features are comprised of several features that are highly correlated in addition to irrelevant features which do not contribute effectively towards the classification process. Furthermore, these redundant features adversely affect the detection efficiency of the IDS as well as increase the computational time, memory and other IT resources [7]. Data pre-processing involves several stages namely data cleaning, data normalization and selection of optimal feature subset from the pertinent feature sets for the given dataset. As mentioned in section 3.1, we have chosen to work with CICIDS - 2017 dataset as it the latest IDS dataset that is publicly available and also represents the real time networks of our times including latest security threat scenarios. CICIDS - 2017 dataset has been collected into eight files and consists of 2,830,743 total records. Each record is described by 78 features. This is a labelled dataset with each record marked as Benign or as one of the attacks out of the fourteen threat categories covered by CICIDS- 2017 dataset. We also describe the importance of feature selection techniques and explain our proposed ensemble based feature selection method.

4.2.1.1 Data Filtration

CICIDS - 2017 dataset is no exception to data redundancy and missing values. These instances can influence the efficiency and accuracy of intrusion detection in a negative manner and hence, it is prudent to remove these records from the dataset. CICIDS - 2017 has eight files which cover the threat scenarios of the dataset and 79 features plus 1 label. The feature 'Fwd Header Length' is repeated in the dataset, hence, one set of occurrence of this feature needs to be removed. Further, the feature 'Flow Packet/s' has several records with inconsistent values such as 'Infinity' and 'NaN' and hence, we need to get rid of these instances from the dataset. In addition to this, there are features with constant values; therefore, these features do not have a substantial contribution to the

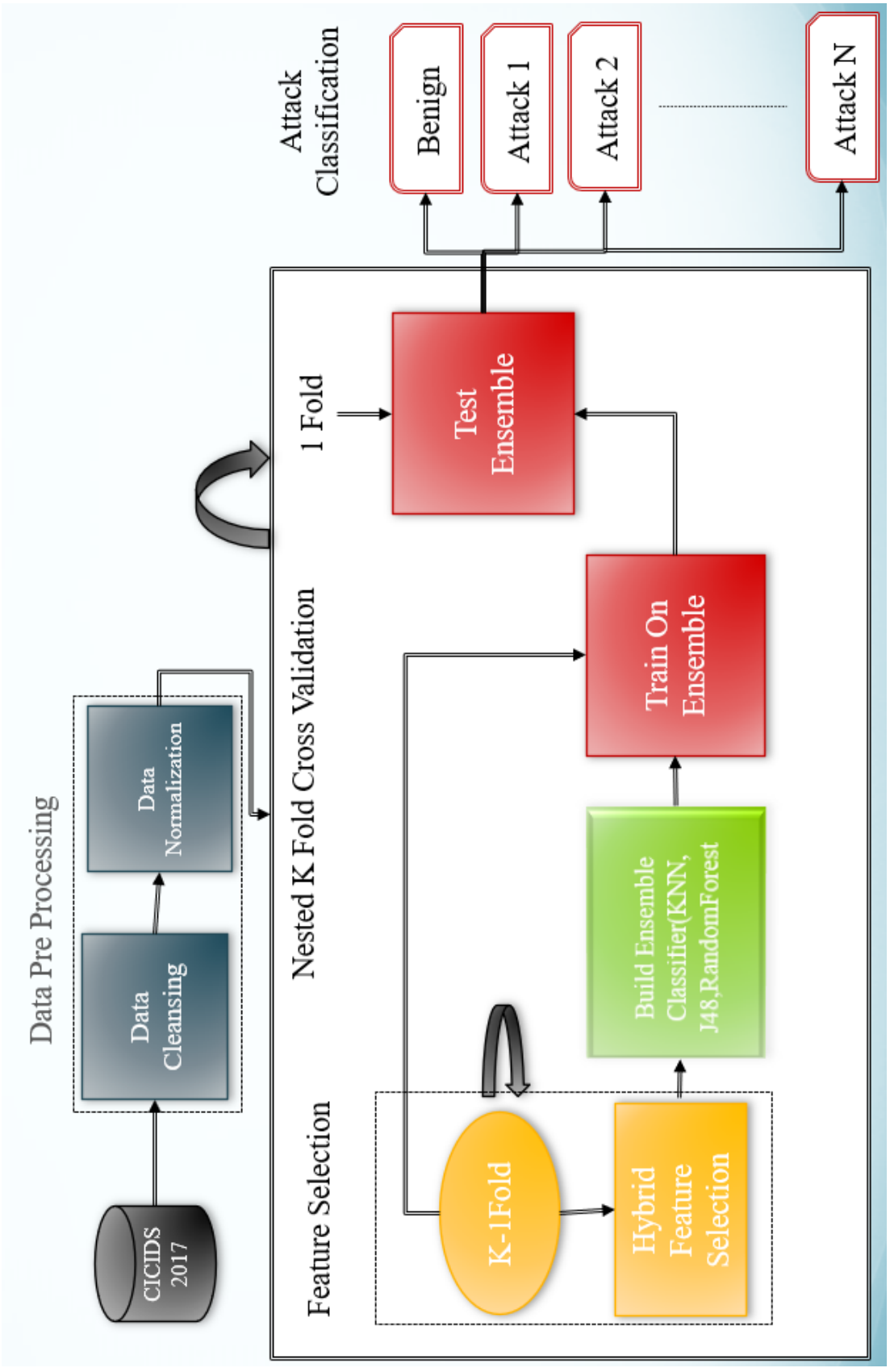

Figure 4.4: Proposed Ensemble Feature Selection and Ensemble-based Network Intrusion Detection Framework

classification process. After data filtering, the CICIDS - 2017 dataset is now reduced to 68 features.

4.2.1.2 Data normalization

Data normalization is usually performed when the dataset has features with values ranging from very high to very low. This variation can lead to a bias towards the larger values and the IDS detection results can be misleading. However, the process of data normalization effectively facilitates in removing this bias by scaling all the attributes for every instance in the dataset in such a way that they fall in the range of [0,1] [8]. We use the minimum-maximum method as mentioned in [97] for data normalization, which is calculated using the following formula :

$$\bar{x} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{4.1}$$

xmin and xmax are the minimum and maximum values for the feature x. Data normalization is done for all the three datasets that we are going to use in our experiments described in Chapter 5.

4.2.2 Proposed Ensemble-based Feature selection technique

The ensemble-based feature selection module in Figure 4.4, uses three machine learning algorithms : IG, CFS and PSO, to generate individual feature subsets. These three individual feature subsets are further combined using Subset Combination Method and an optimal ensemble-based feature subset for the CICIDS - 2017 dataset. This proposed FS scheme has been explained in Chapter 3, section 3.4. The implementation of this feature selection framework and the results for the same, are further described in chapter 5, section 5.1.2

4.2.3 Proposed ensemble-based IDS framework explained

The decision engine in our IDS framework is implemented using the ensemble-based ML algorithm. The base classifier, KNN, when used as a single classifier for prediction of anomaly detection using CICIDS - 2017 dataset, is not very effective in predicting all the attack categories presented by the CICIDS - 2017 dataset. Hence, to improve

the performance of this weak single prediction classifier, we propose an ensemble-based intrusion detection approach that employs three machine learning algorithms namely K-Nearest Neighbors (KNN), C4.5/J48 and Random Forest (RF) as base learner classifiers for building the ensemble classifier. Our ensemble-based IDS model, thus is able to enhance the detection performance of IDS scheme. Before we proceed to explain the proposed IDS framework, a brief theoretical background for the base classifiers used in our scheme, is given below :

1. K-Nearest Neighbors (KNN): K - Nearest Neighbor (KNN) machine learning method falls under non-parametric classifier. KNN is based on the assumption that features with similar properties will be found close together. This is achieved by measuring the distance between points on the graph. Euclidean distance is the most commonly used method. The aim of this method is to predict the label for a data sample based on the class that is closest to the K nearest neighbor. As shown in Figure 4.5, the value of K is 5 and X is the point that needs to be classified. Looking at the figure we can see that out of the five nearest neighbors for the point X, three are anomalies and only two instances are normal. Hence, point X is classified as an Anomaly or Intrusion. This is the basis of classification for K-Nearest Neighbor (KNN) based algorithms when used in designing IDS. Choice of K is crucial and literature shows that an odd value of K is chosen [90].

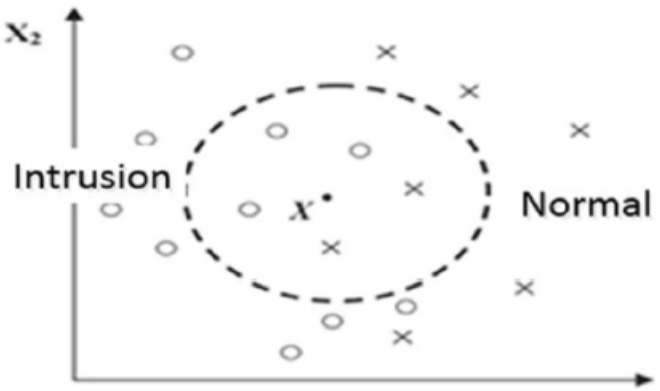

Figure 4.5: Classification example of KNN for K=5 [90]

2. C4.5: it comes under the family of Decision Tree (DT) the Iterative Dichotomiser (ID3) [5]. ID3 algorithm was first developed by Ross Quinlan, who further formu-

lated C4.5 algorithm which is also called J48 and is a widely used algorithm in literature [138]. The pseudo code for C4.5/J48 algorithm is given in Figure 4.6. Unlike ID3, the input for C4.5 algorithm can be both continuous and categorical features. Top down or bottom up techniques can be employed to build the C4.5 DT [5].

```
1: Create a root node N;
2: IF (T belongs to same category C)
       {leaf node = N;
         Mark N as class C;
         Return N;
       }
3: For i=1 to n
       {Calculate Information_gain  (Ai);}
4: ta= testing attribute;
5: N.ta = attribute having highest information_gain;
6: if (N.ta == continuous )
        { find threshold;}
7: For (Each T in splitting of T)
8:       if (T is empty)
            {child of N is a leaf node;}
            else
            {child of N= dtree T)}
10: calculate classification error rate of node N;
11: return N;
```

Figure 4.6: C4.5(J48) pseudo code [5]

3. Random Forest : As the name suggests, Random Forests are a group of trees which are used for making predictions and the final outcome is obtained by combining the individual tree predictions. These schemes need tuning parameters namely *mtry* and *ntree* where *mtry* is the number of attributes for every split and these attributes are randomly sampled. *ntree* is the total number of trees. [34]. Each tree in the Random Forest algorithm is tested using a data subset of the original

training data. This subset is generated using a bootstrapping technique. As a result of using the bootstrapped generated data subsets, only two-thirds of the total training data is used by each tree. The remaining one-third data is known as Out Of Bag (OOB) data which can further be used to validate the model performance. Thus, Random Forests are a type of ensemble algorithm. The advantages of this algorithm is that it is flexible, has high rate of prediction and does not suffer the over-fitting issue on the basis of tree numbers being examined. [140].

4.2.3.1 Vote

Vote algorithm is used to combine the individual base learners, KNN, C4.5 and Random Forest, and their individual predictions using a combination rule. Minimum probability, maximum probability, majority voting, product of probabilities, and average of probabilities are combination rules that can be applied with voting algorithm. However, as pointed out by the authors in [193], for multi-class classification, average of Probabilities (AOP) is used since the number of predicted classes is larger than the number of base classifiers [193].

4.2.3.2 K-Fold Cross Validation

Several studies in literature use the holdout method for create training and testing datasets by splitting the original data into a 70-30 % or 80-20 % split. This may not be the best approach for testing the model performance, since there is a high possibility of relevant information being ignored during the creation of training dataset and the same may be available in the testing dataset. Hence, this may introduce bias in the performance accuracy of the prediction model. K- fold cross validation technique involves randomly splitting the dataset samples into K number of fixed data partitions, and use K-1 partitions for training and the last remaining Kth partition for testing the prediction accuracy of the proposed IDS model. This process is repeated every partition has been used as testing dataset. Once the iterative process is completed, the average of predicted accuracy is calculated to get the final performance accuracy of the proposed model. K-Fold cross validation technique is effective in reducing over-fitting issue for machine learning prediction models. We use K-fold [158] cross validation technique for ensemble-based feature selection and also for training and testing of the ensemble-based intrusion detection model shown in Figure 4.4, where the value of K = 10 is chosen

empirically as found in literature.

We have used the latest publicly available dataset CICIDS -2017 to evaluate the performance of our proposed model. As explained in Chapter 3, Section 3.3, we have derived ensemble-based feature subset for CICIDS -2017 using our proposed ensemble based feature selection method, as shown in Figure 3.18. In addition, we have used NSL - KDD IDS dataset to compare the performance of our model. The proposed ensemble-based IDS model has been implemented for CICIDS - 2017 and NSL - KDD datasets with original features and ensemble-based selected features that have been generated using our proposed ensemble-based FS method. K-fold cross validation is used for generating the ensemble-based features using our proposed FS method. These selected ensemble-based features are used as input for the proposed ensemble-based IDS scheme where the three base classifiers, KNN, c 4.5 and Random Forest, are used to predict the attack class individually using K-fold cross validation. Vote algorithm using average of probabilities as the combining rule is further used to combine these individual predictions into a single ensemble-based classifier output providing the final attack class prediction. The proposed ensemble-based IDS model is used to classify multi-class attacks in each of these datasets. These experimental implementation steps and performance evaluation of our proposed scheme are explained in elaborate detail in chapter 5.

4.3 Chapter Summary

In this chapter we address the second problem statement mentioned in Chapter 1, Section 1.2. The drawback of a "weak" ML classifier results in IDS model performance which is not very impressive. Further, it is also not possible to chose one ML algorithm as the best out of the many available as each method is used for a specific anomaly classification problem. Single classifier ML schemes also suffer from the issue of over-fitting. The advantage of using ensemble based intrusion detection schemes is that the ensemble classification approach addresses all the aforementioned issues with single classifier ML techniques. There are a few disadvantages, such as classifier learning process can get quite complicated in learning which may impact the initial classifier training time. These methods can be more expensive than single ML methods. However, the advantages of ensemble- based methods certainly outweigh the disadvantages making them one of the most researched techniques for attack classification within the IDS literature. We

have presented our ensemble-based intrusion detection scheme which uses the 10-fold cross validation method to generate ensemble-based feature subset and also to train and test the proposed ensemble-based IDS model. The ensemble classifier is generated from the base classifiers namely, KNN, C4.5/J48 and Random Forest using the voting meta-algorithm and Average of Probabilities (AOP) method is used as the combination rule to combine the individual predicted probabilities from the single base classifiers and provide the final model prediction accuracy.

On the basis of our ensemble-based literature review shown in Table 4.1 , we come to the conclusion that Ensemble based intrusion detection models have been popular with the intrusion detection research community for the last decade. However, we also note that the majority of the ensemble based research work does not employ any feature selection to reduce the dimensionality of the input datasets. Further, we observe that most of the studies have used majority voting method for combining base ensemble classifiers. However, it is quite evident from our research, that the lack of availability of modern day validation IDS datasets, is a matter of concern for the research community as it still uses age-old IDS datasets for IDS model performance evaluation. The recently published IDS datasets namely, CICIDS-2017 is not being widely used by researchers yet. Unavailability of benchmark studies using these new datasets can be one of the reasons for this. Hence, the need for bench-marking the new dataset with feature selection data subsets and it is important for for future researchers to start using CICIDS - 2017. Chapter 5 shows the implementation results and analysis for the proposed framework.

CHAPTER 5

RESULTS AND ANALYSIS

We presented the proposed ensemble-based feature selection scheme in Chapter 3 and the proposed ensemble-based intrusion detection scheme using ML base classifiers namely, KNN, C4.5 and Random Forest, in Chapter 4. Chapter 3 and Chapter 4 address the first and second problem statement as identified in Chapter 1, Section 1.2. In Chapter 5 we explain the experimental design and implementation for the proposed feature selection and intrusion detection framework as shown in detail in Figure 4.4 in Chapter 4, Section 4.2. CICIDS - 2017 dataset is used for performance evaluation of the proposed scheme. CICIDS - 2017 data pre-processing and feature selection are explained with implementation steps. In addition, the results of the implementation of the proposed model is shown in detail in this chapter. We also present the performance results for our proposed framework using CICIDS - 2017 for feature selection and intrusion detection accuracy.

Section 5.1 provides the results and analysis, obtained by implementing the Data pre-processing, proposed ensemble-based feature selection and proposed ensemble-based intrusion detection scheme using CICIDS-2017 dataset. We also present a comparative analysis with existing schemes in literature that use CICIDS - 2017 dataset for IDS model validation. Section 5.2 provides the results for the implementation of our proposed framework for NSL - KDD dataset, in order to provide benchmark comparison for the results obtained by using the same framework with CICIDS - 2017 dataset.

5.1 Results from the implementation of Proposed Framework with CICIDS - 2017 Dataset

In the following sections, the results obtained from implementation of the proposed framework is explained for CICIDS - 2017 dataset. The performance of the framework using the CICIDS - 2017 dataset is further compared by implementing the framework with NSL - KDD dataset.

5.1.1 CICIDS - 2017 Data Filtering Process

As the CICIDS - 2017 dataset was generated by capturing network data from diverse sources, it consists of noisy and redundant data, which needs to be filtered in the beginning to avoid a dismissive impact on the detection accuracy. We use data filtering step to remove features such as *'Fwd Header Length'* which is repeated in the original dataset. Also features *'Bwd PSH Flags'*, *'Bwd URG Flags'*, *'Fwd Avg Bytes/Bulk'*, *'Fwd Avg Packets/Bulk'*, *'Fwd Avg Bulk Rate'*, *'Bwd Avg Bytes/Bulk'*, *'Bwd Avg Packets/Bulk'*, *'Bwd Avg Bulk Rate'*, *'Fwd URG Flag'* and *'CWE Flag Count'* have a value of zero and hence not included. The original CICIDS - 2017 dataset has 78 original features in addition to the label. After data cleansing and removing the redundant features, we are able to reduce the features to 68. In addition there are multiple records with 'NaN' and 'Infinity' values and hence, these need to be removed. The original dataset before data filtration consists of 2,827,876 records which is reduced to 1,666,532 records after performing the above mentioned data filtration. This is still a very large number of records and processing all of them would have been a huge computational overhead. Hence, we sample the CICIDS - 2017 dataset to generate approximately 30 percent of the original dataset which is a good representation of the original CICIDS - 2017 dataset and saves testing and training of the proposed ensemble model. The details of the records for each class in the sampled CICIDS - 2017 are shown in Table 5.1. However, the 30 percent sampled dataset still consists of a large feature number.

In addition to data filtration, values for some features in the dataset have large variation. To fix this, the sampled data is normalised using the maximum-minimum method for data normalisation [97]. The data values for all instances fall in the range of [0,1].

Class	Total Records
Benign	188,955
Bot	1,956
DDoS	99,999
Portscan	158,800
Web Attack BruteForce	1,507
Web Attack - XSS	652
Web Attack - SQL Injection	21
FTP - Patator	7,935
SSH - Patator	5,897
DoS Slowloris	5,796
DoS Slowhttptest	5,499
Dos Hulk	88,704
DoS Goldeneye	10,293
Heartbleed	11
Infiltration	36
Total Records	576,061

Table 5.1: Record details for CICIDS - 2017 after Data Filtering

5.1.2 CICIDS - 2017 Ensemble-based Feature Selection Process

The importance of feature selection has been explained in detail in Chapter 3. We explain the process of generating ensemble- based features using our proposed ensemble feature selection scheme for CICIDS-2017 dataset below. This approach is illustrated in Figure 3.18. As described earlier in Section 3.3 of Chapter 3, our proposed feature selection method uses three FS algorithms : Information Gain (IG), Correlation Feature Selection (CFS) and Particle Swarm Optimization (PSO). Each of these methods generates a subset of best possible features using the sample dataset which was created after data filtration and data normalization in the previous section 5.1.1.

Figure 5.1 lists the individual list of features generated for Information Gain, Correlation and Particle Swarm Optimization FS methods. The number of individual features for IG, CFS and PSO methods is 27, 21 and 13 respectively. For the Information Gain method, features with a rank value greater than 1 were selected while for Correlation Based Feature method the cut off value was 0.2. Further, the Subset Combination method (SCM) [26] was employed to generate 17 ensemble features from these three feature subsets. The selected ensemble features based on our proposed feature selection method are listed in Table 5.2. These 17 chosen ensemble-based features provide a strong

representation of the CICIDS - 2017 dataset and we will use these for our proposed ensemble-based intrusion detection scheme. 10-Fold cross validation technique is applied for feature selection, where the dataset is randomly split into 10 parts. For each iteration, 9 out of the 10 parts are used for training and 1 part is used for testing, thus ensuring that every subset has equal possibility to be used for testing and training [74].

Feature Number	**Feature Name**
1	Destination Port
7	Fwd Packet Length Max
11	Bwd Packet Length Max
12	Bwd Packet Length Min
13	Bwd Packet Length Mean
18	Flow IAT Std
19	Flow IAT Max
24	Fwd IAT Max
37	Max Packet Length
38	Packet Length Mean
39	Packet Length Std
40	Packet Length Variance
49	Average Packet Size
51	Avg Bwd Segment Size
56	Init_Win_bytes_forward
57	Init_Win_bytes_backward
59	min_seg_size_forward

Table 5.2: CICIDS - 2017 Selected Ensemble Features after Data Filtering

Table 5.3 shows the comparison between the three individual FS methods and the ensemble features. It can be observed that though the performance of the model with 27 IG features and with 17 Ensemble features is the same , it is important point out here, that the number of features for IG is much larger than ensemble based features (17). Hence, we have achieved the same accuracy using the ensemble-based feature selection which has resulted in reduced number of features for CICIDS - 2017 dataset as compared to individual feature selection algorithms (IG, CFS and PSO). Thus we claim our ensemble based FS approach performs better than the standalone individual feature selection schemes.

Infogain	Correlation	CFS PSO
Feature Name	**Feature Name**	**Feature Name**
Average Packet Size	Bwd Packet Length Mean	Destination Port
Packet Length Mean	Avg Bwd Segment Size	Total Length of Bwd Packets
Packet Length Std	Packet Length Std	Fwd Packet Length Max
Bwd Packet Length Mean	Packet Length Mean	Bwd Packet Length Max
Avg Bwd Segment Size	PSH Flag Count	Bwd Packet Length Min
Fwd Packet Length Max	Bwd Packet Length Max	Flow IAT Max
Total Length of Fwd Packets	Average Packet Size	Flow IAT Min
Subflow Fwd Bytes	Max Packet Length	Avg Bwd Segment Size
Init_Win_bytes_forward	Bwd Packet Length Std	Subflow Bwd Bytes
Max Packet Length	Packet Length Variance	Init_Win_bytes_forward
Bwd Packet Length Max	Min Packet Length	Init_Win_bytes_backward
Init_Win_bytes_backward	Bwd Packet Length Min	act_data_pkt_fwd
Fwd Packet Length Mean	Fwd IAT Std	min_seg_size_forward
Avg Fwd Segment Size	min_seg_size_forward	
Destination Port	Flow IAT Max	
Bwd Header Length	Fwd IAT Max	
Packet Length Variance	Idle Max	
Fwd Header Length	Idle Mean	
Flow Duration	Idle Min	
Flow IAT Max	Flow IAT Std	
Fwd IAT Max	ACK Flag Count	
Fwd IAT Total		
Flow IAT Mean		
Flow IAT Std		
Fwd IAT Mean		
Total Backward Packets		
Subflow Bwd Packets		

Figure 5.1: Individual features generated for IG, CFS and PSO

FS Method	Acc	Precision	Recall	F-Measure	MCC
IG - 27 Features	0.998	0.998	0.998	0.998	0.998
CFS - 21 Features	0.983	0.983	0.983	0.982	0.978
PSO - 13 Features	0.997	0.998	0.997	0.997	0.997
Ensemble - 17 Features	0.998	0.998	0.998	0.998	0.998

Table 5.3: Comparative results of Feature Selection methods for CICIDS - 2017

5.1.3 Results from implementation of Proposed Ensemble-based intrusion detection model with CICIDS - 2017

The implementation of the proposed ensemble model as shown in Figure 3.18 is explained in this section. It has been mentioned in Chapter 4, that our ensemble method uses KNN, C4.5 and Random Forest algorithms as base classifiers. With the KNN algorithm, the value of the parameter K is chosen as 20. This value for K was chosen after running several iterations of the KNN algorithm with CICIDS-2017 dataset, with the aim to finding the value of K, for which the classification accuracy is best.

Further, as we have explained in section 4.2.2, of chapter 4, the proposed ensemble-based IDS scheme combines the three base classifiers (KNN, C4,5 and Random Forest algorithm) using voting method and average of probabilities (AOP) combination rule to calculate the final classification outcome of the ensemble-based IDS model for the CICIDS - 2017 dataset. 10-Fold cross validation is used for training and testing the proposed ensemble method.

Table 5.4 and Table 5.5 present the overall performance of the Ensemble model for the CICIDS - 2017 dataset with the original feature set of 68 features and the reduced feature set of 17 features generated using our proposed ensemble FS method. We observe from the numbers in both these tables, that the accuracy for our ensemble model is the same as that for the three base classifiers. Thus, the proposed ensemble based IDS scheme with ensemble-based feature selection technique (17 ensemble features), has achieved high performance results using the CICIDS - 2017 dataset. These results are as good as those obtained when the complete feature set (68 features) was used with our proposed scheme. This observation leads us to the conclusion that using our ensemble-based feature selection algorithm, we have reduced the high dimensional CICIDS - 2017 database without compromising on the performance parameters.

However, we note that when KNN is used as single classifier for the IDS model, precision, recall, F-Measure and MCC values quite low as compared to our proposed ensemble-based IDS framework. This is due to the minority class (SQL Injection, Heartbleed and Infiltration attacks) present in the CICIDS - 2017 dataset which are not detected by KNN. The proposed ensemble-based IDS model, however, is successful in

Classifier	Acc	Precision	Recall	F-Measure	MCC
C4.5	0.998	0.998	0.998	0.998	0.998
Random Forest	0.998	0.998	0.998	0.998	0.998
KNN	0.998	0.830	0.995	0.879	0.885
Ensemble	0.998	0.998	0.998	0.998	0.998

Table 5.4: Overall Performance of Ensemble model with CICIDS2017 Dataset (68 features)

classifying the minority classes and has Precision, Recall, F - Measure and MCC values which are improved when compared to KNN. Thus, accuracy should not be considered as the only measure for performance evaluation of an IDS as it can be misleading. Precision, Recall, F-Measure and MCC are equally important performance evaluation metrics which should be used in addition to the Accuracy value for evaluating the IDS scheme. These metrics have been explained in detail in section 2.5 of Chapter 2. It can be observed that the proposed ensemble-based scheme is comparable with the individual base learners, C4.5 and Random Forest algorithms. However, the precision for KNN algorithm is comparatively low when compared to ensemble-based method, thus indicating that using our proposed method has led to improvement of precision value for the base classifier KNN.

Classifier	Acc	Precision	Recall	F-Measure	MCC
C4.5	0.998	0.998	0.998	0.998	0.997
Random Forest	0.998	0.998	0.998	0.998	0.998
KNN	0.996	0.847	0.996	0.828	0.837
Ensemble	0.998	0.998	0.998	0.998	0.998

Table 5.5: Overall Performance of Ensemble model with CICIDS2017 (17 features)

Table 5.6 presents the performance of the proposed ensemble scheme for each class in the CICIDS - 2017 dataset with the 17 ensemble generated features. As can be seen, the proposed ensemble model performs best for DDoS attacks, Dos attacks (Slowloris, Slowhttptest, Hulk, GoldenEye and Heartbleed), Portscan attack and Web Attacks (FTP, SSH and Brute Force). However the classification accuracy for SQL Injection and Cross-scripting (XSS) attacks is low. The overall performance accuracy for the proposed ensemble intrusion detection scheme with 17 ensemble features is 99.807 percent.

Class Name	Acc	Precision	Recall	F-Measure	MCC	ROC
Benign	0.999	0.999	0.999	0.999	0.999	1.000
Bot	0.989	0.996	0.989	0.992	0.992	1.000
DDoS	1.000	1.000	1.000	1.000	1.000	1.000
Portscan	1.000	1.000	1.000	1.000	1.000	1.000
Web Attack BruteForce	0.838	0.678	0.838	0.749	0.753	0.999
Web Attack - XSS	0.281	0.425	0.281	0.338	0.345	0.995
Web Attack - SQL Injection	0.333	0.875	0.333	0.483	0.540	0.975
FTP - Patator	0.999	1.000	0.999	1.000	1.000	1.000
SSH - Patator	1.000	1.000	1.000	1.000	1.000	1.000
DoS Slowloris	0.995	0.997	0.995	0.996	0.996	0.999
DoS Slowhttptest	0.994	0.996	0.994	0.995	0.995	1.000
Dos Hulk	1.000	0.999	1.000	1.000	1.000	1.000
DoS GoldenEye	0.998	0.998	0.998	0.998	0.998	1.000
Heartbleed	0.909	1.000	0.909	0.952	0.953	1.000
Infiltration	0.722	1.000	0.722	0.839	0.850	0.986

Table 5.6: Attack Class Performance of Ensemble model for CICIDS2017 (17 features)

Further, in the Appendix section of this thesis, we have also presented the ROC graphs (Figure A.1 to A.15) for each attack class listed in Table 5.6.

Comparison of our ensemble model with similar literature studies that are using CICIDS - 2017 dataset for multi-class detection is shown in Table 5.7. As can be seen, our proposed ensemble scheme performs equally well as the existing methods used for intrusion detection in literature. We make a note here, that since CICIDS - 2017 is a relatively newly published IDS dataset, there are not many IDS research works that use it for IDS performance validation. Some researches [13] [20] [60] [154] [192], [194], have used CICIDS - 2017 dataset for intrusion detection schemes. However, they have only focused on classifying the DoS attack family in the CICDS - 2017. Thus, we claim that the research work presented in this thesis provides a complete evaluation of the CICIDS - 2017 dataset using ensemble-based feature selection method and an ensemble-based intrusion detection model for network anomaly classification.

Study	Detection Method	FS	Acc	Precision	Recall	F1 Measure
Ferrag et al. [62]	DNN	–	0.9728	–	–	–
	RNN	–	0.9731	–	–	–
	CNN	–	0.9376	–	–	–
	RBM	–	0.9728	–	–	–
	DBN	–	0.9730	–	–	–
	RNN	–	0.9731	–	–	–
	RNN	–	0.9737	–	–	–
Kanimozhi et al. [87]	ANN	–	0.9997	0.9996	1.000	0.9998
	RF	–	0.9983	0.9992	0.9988	0.9992
	KNN	–	0.9973	0.998	0.9988	0.9984
	SVM	–	0.998	0.9	0.9988	0.9994
	Adaboost	–	0.9996	0.9996	0.9988	0.9992
	NB	–	0.992	0.9929	0.9976	0.9953
Proposed Model	Ensemble	Ensemble	0.998	0.998	0.998	0.998

Table 5.7: Comparison of proposed model with existing schemes using CICIDS - 2017

5.2 Feature Selection and Ensemble-based Detection for NSL - KDD dataset

We further implemented the proposed framework shown in Figure 3.18 with the traditional IDS dataset namely NSL - KDD dataset with the aim to evaluate the model. As mentioned in Chapter 3, NSL - KDD dataset has 41 features in addition to 1 class label and 148,516 records for training and testing data. It covers 5 attack classes namely DoS, U2R, R2L and Probe and a normal class. Figure 5.2 gives the details of the classes available.

After initial data pre-processing of NSL - KDD dataset to represent the five classes as real values, as shown in [61] [31], we implement the proposed feature selection scheme described before. Feature selection is performed using 10-Fold cross validation method which is also explained in the previous section of this chapter. Table 5.8 gives the list of ensemble features for NSL - KDD.

Class Name	Number of records
DoS	53,383
U2R	254
R2l	3,749
Probe	14,077
Normal	77,053

Figure 5.2: Class distribution for NSL - KDD dataset

Feature Number	**Feature Name**
3	service
4	flag
6	dst_bytes
12	logged_in
14	root_shell
22	is_guest_login
26	srv_serror_rate
29	same_srv_rate
30	diff_srv_rate
33	dst_host_srv_count
34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate
37	dst_host_srv_diff_host_rate
39	dst_host_srv_serror_rate

Table 5.8: NSL - KDD Selected Ensemble Features

The proposed ensemble-based IDS model is implemented using NSL - KDD dataset. The overall performance classification of the proposed ensemble scheme on NSL - KDD dataset using the complete feature set (41 features) is shown in Table 5.9. Table 5.10 shows the overall performance results obtained when the ensemble generated 15 features were used for the validation of proposed ensemble-based IDS scheme. As seen in the obtained results, the accuracy for the proposed ensemble-based IDS scheme using ensemble-based feature selection method (15 features) for NSL - KDD dataset is 0.980 which is quite similar to that obtained by the same IDS model when no feature selection is implemented and the complete feature set for NSL - KDD dataset (41 features) are

used. Thus it is safe to claim, that the IDS detection accuracy for the proposed ensemble-based IDS scheme, provides comparable detection accuracy with reduced, better quality informative ensemble features subset. Further, the classification performance for each attack class in NSL - KDD dataset using the ensemble-based feature selection (15 features) is presented in Table 5.11. Further, in the Appendix section of this thesis, we have also presented the ROC graphs (Figure A.16 to A.20) for each attack class listed in Table 5.11.

Classifier	**Acc**	**Precision**	**Recall**	**F-Measure**	**MCC**
Ensemble	0.988	0.988	0.988	0.988	0.981

Table 5.9: Overall Performance of Ensemble model with NSL-KDD dataset (41 features)

Classifier	**Acc**	**Precision**	**Recall**	**F-Measure**	**MCC**
Ensemble	0.980	0.980	0.980	0.980	0.968

Table 5.10: Overall Performance of Ensemble model with NSL-KDD dataset (15 features)

Class Name	**Acc**	**Precision**	**Recall**	**F-Measure**	**MCC**	**ROC**
DoS	0.993	0.987	0.993	0.990	0.984	1.000
U2R	0.394	0.971	0.394	0.560	0.618	0.995
R2L	0.811	0.898	0.811	0.852	0.850	0.997
Probe	0.955	0.971	0.955	0.963	0.959	0.999
Normal	0.986	0.981	0.986	0.984	0.966	0.999

Table 5.11: Attack Class Performance of Ensemble model with NSL-KDD dataset (15 features)

In addition to CICIDS - 2017 and NSL - KDD dataset, we implemented the proposed Ensemble-based model for binary classification with UNSW-NB15 dataset [32] for binary classification of attacks. Table 5.12 and 5.13 show the overall performance of the proposed detection method with original dataset features (44 features) and ensemble features (24 features) respectively for binary classification of attacks.

Classifier	Acc	Precision	Recall	F-Measure	MCC
Ensemble	0.987	0.987	0.987	0.987	0.972

Table 5.12: Overall Performance of Ensemble model with UNSW-NB15 dataset (44 features)

Classifier	Acc	Precision	Recall	F-Measure	MCC
Ensemble	0.985	0.985	0.985	0.985	0.968

Table 5.13: Overall Performance of Ensemble model with UNSW-NB15 dataset (24 features)

Finally we note the model generation time for CICIDS - 2017 as shown in Table 5.14. The time taken by the IDS model with no feature selection is much higher, which is expected due to the computational overheard because of the high dimensionality of CICIDS - 2017 dataset. However, the IDS model takes nearly half the time when we implement feature selection. This is one of the several advantages of feature selection as it improves the time taken for detection of intrusions.

Without Feature Selection	With Ensemble Feature Selection
839.63 seconds	466.78 seconds

Table 5.14: Comparison of the proposed scheme based on model generation time

5.3 Chapter Summary

The performance evaluation of the proposed ensemble model was presented in this chapter. The process of ensemble feature selection and ensemble IDS detection was explained using the latest IDS dataset : CICIDS - 2017. Comparison of model performance and model build time indicates a few important conclusions. Single classifier based IDS using KNN with no feature selection for CICIDS - 2017 dataset fails to classify minority attacks like Heartbleed, SQL injection and Infiltration for the dataset. However, ensemble based IDS model with no Feature Selection is successful in classifying these minority attacks in an acceptable range. The accuracy for ensemble model with no feature selection is 99.835 percent which is very good and better than the accuracy of single classifier KNN IDS model. However, ensemble model build time is higher than single classifier IDS

model. However, this build time value is again not very large and hence, acceptable.

In general, when ensemble based IDS models were implemented with feature selection, the model build time was much lower. In particular, the time for the ensemble based model with ensemble features is reduced by 55 percent when compared to ensemble model with no FS.

Proposed Ensemble IDS with 17 Hybrid Features demonstrates 99.81 percent accuracy

Model	Build Time (sec)	Accuracy (%)
KNN No FS	0.14	99.55
Ensemble No FS	839.63	99.835
Ensemble with Correlation Features (21 FS)	392.5	98.27
Ensemble with InfoGain Features (27)	459.878	99.83
Ensemble with CFS PSO Features (13)	742.25	99.75
Ensemble Model with Ensemble Features (17)	466.78	99.81

Figure 5.3: Proposed Feature Selection and Ensemble Model summary

which is nearly the same as Ensemble based IDS with no Feature Selection (68 features). Additionally the model build time for our proposed scheme is 466.78 sec which is 55.59 percent less time taken in comparison to the model build time with CICIDS - 2017 dataset with original features. Figure 5.3 provides the performance of the proposed ensemble-based IDS scheme using CICIDS - 2017 dataset, in a snapshot. Further, the performance of the proposed model with NSL - KDD dataset was very impressive with accuracy of 98.0231 percent.

Lastly, for the CICIDS - 2017 dataset, the ensemble model with feature selection has an accuracy which matches that of the ensemble model with no feature selection performed. This is a valuable contribution to the field of intrusion detection as using our proposed feature selection method we reduced the dimensionality of the CICIDS - 2017 dataset. However, even with low number of features (17) as compared to the complete feature set (68) for CICIDS - 2017 dataset, the ensemble model performance has not been

impacted. This demonstrates that the ensemble features generated using the proposed FS scheme in this thesis, generates an optimum and highly informative feature subset. With this, we conclude that the proposed Ensemble IDS with Ensemble Feature selection method addresses the challenges from 2-5 that are mentioned in Chapter 1, Section 1.2 .

CHAPTER 6

CONCLUSION AND FUTURE WORKS

Intrusion detection is an area of cybersecurity that attracts researchers who are motivated to develop novel intrusion detection schemes to detect and prevent sophisticated attacks.

6.1 Thesis Contributions

The contributions made by this dissertation are recapped here:

We have uncovered three main issues in the field of intrusion detection which are primarily related to :

1. Lack of use of ensemble methods for feature selection problem related to the high dimensionality IDS datasets.
2. Single classifier machine learning algorithms which lead to a weak classifier and hence, impact the IDS model performance adversely.
3. Over usage of traditional bench mark IDS datasets like DARPA 98 and KDD 99, in the absence of a modern publicly available IDS validation datasets which may lead to misleading IDS performance evaluation, since the ancient datasets do not contain recent attacks or the modern network topology.

We have addressed these gaps in existing IDS research as follows :

1. We have proposed, developed and implemented an ensemble base feature selection scheme which is used to combine the three individual feature selection techniques : Information Gain, Correlation Feature Selection and Particle Swarm Optimization. We have shown through our experimental results using the proposed ensemble scheme, that the 17 features generated by our ensemble FS method give the same high accuracy as each of the individual FS methods which use a higher number of features compared to our scheme. Thus, the IDS model generated using the ensemble features has lower computational time as compared to single FS approaches. Further, the importance of ensemble based feature selection techniques for dimensionality reduction is reinforced on the basis of the experimental result shown in this thesis. This helps in giving a resolution for the first identified research gap.

2. We have proposed, developed and implemented an ensemble based intrusion detection model to enhance the detection accuracy of the single ML classifier - KNN which does not perform well in detecting the minority classes for CICIDS - 2017 dataset. Thus, with our ensemble IDS model, we use three base classifiers : KNN, C4.5 and Random Forest, to build an ensemble classifier which is capable of better multi-class attack classifications for the CICIDS - 2017 dataset. This contribution addresses the second issue of single ML classifiers that have a weak performance for intrusion detection.

3. We have chosen the latest available CICIDS - 2017 dataset that represents the network setup of current times and also covers the latest sophisticated network threats, thus making it an ideal IDS performance evaluation dataset. However, the lack of benchmark features and IDS performance with ML algorithms with CICIDS - 2017 dataset has led to few researchers using it. In our literature review, which is part of this dissertation, we have discovered that in spite of being in the year 2020, the majority of the IDS research community is still using KDD 99 and NSL - KDD datasets to benchmark their proposed anomaly detection models. The traditional IDS validation datasets were published more than two decades ago. Our experimental results calculated the accuracy, precision, recall, MCC and ROC values using CICIDS - 2017 dataset to validate the performance of our proposed ensemble IDS model. The performance of CICIDS - 2017 dataset with feature selection using ensemble approach, is very promising and comparable to the

detection accuracy in existing IDS literature. Hence, this thesis makes a valuable contribution step in the direction to make CICIDS - 2017 a bench mark IDS for future IDS research performance evaluation. Thus the third gap is addressed successfully by the work in this thesis.

6.2 Future Work

Intrusion detection is not only limited to traditional networks but, also, plays a significant role in modern enterprise networks like the cloud computing networks. As a part of our literature review, we investigated the collaborative intrusion detection in the cloud. We concluded that the collaborative IDS with alert correlation, offers better cloud security than stand-alone IDS. In addition, we further proposed a collaborative intrusion detection framework for cloud. We propose developing our ensemble-based intrusion detection approach in the cloud architecture, as an extension of our current proposed framework for the future. The high classification accuracy of ensemble classifiers compared to single ML algorithm for IDS detection in the cloud shows considerable promise.

As mentioned throughout our dissertation, CICIDS- 2017 dataset is a comparatively new publicly available intrusion detection dataset. There are several advantages in using CICIDS - 2017, as it represents the latest network threats and modern network systems. Hence, using this dataset helps in evaluation IDS schemes, that are relevant for intrusions in current times and will contribute in developing efficient and robust IDS framework. Thus, as future work we propose using CICIDS - 2017 dataset for IDS scheme evaluation and validation for future research schemes.

In our dissertation, we have successfully used the ensemble method with three feature selection techniques for the feature selection process for dimensionality reduction of CICIDS - 2017 dataset. The approach presented in this thesis is robust and uses the filter-based techniques namely Information Gain, Correlation Feature Selection and Particle Swarm Optimization. We believe that in future there is a lot of scope to investigate ensemble-based schemes for CICIDS - 2017 dataset and aim to create approaches that are robust and generate feature ranking for better feature subsets of the chosen IDS dataset. An ensemble approach using a combination of filter-based methods and embedded methods for feature selection can be another area for further investigation in future, to construct optimized feature subsets for CICIDS - 2017 dataset.

In addition, with an ensemble-based intrusion detection scheme, the choice of base classifiers is very significant. As seen in the literature review presented in this dissertation, many ML classifiers have been considered for ensemble-based intrusion detection schemes by different researchers. There is a big scope of research in choosing ML base classifiers for future ensemble-based intrusion detection models. It will be interesting to see research that generates a robust and efficient ensemble classifier using new combinations of base classifiers that have not been investigated yet, such as the meta-heuristic machine algorithms. Hence, the works presented in this dissertation pave ways to apply and investigate further in IDSs.

APPENDIX

A.0.1 ROC graphs generated for Benign and attack classes of the CICIDS - 2017 dataset and the NSL - KDD dataset

The ROC values for our proposed ensemble-based feature selection and ensemble-based IDS framework for the CICIDS - 2017 dataset have been listed in Table 5.6. The ROC graphs for Benign and each Attack class of the CICIDS - 2017 dataset as shown in the Table 5.6 are represented in Figure A.1 to Figure A.15.

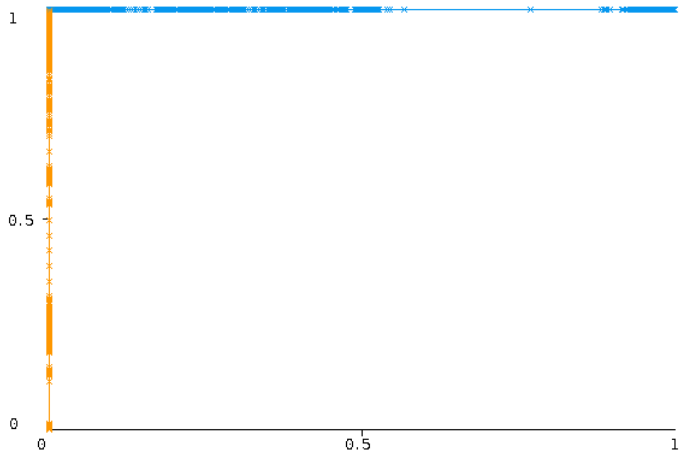

Figure A.1: CICIDS - 2017 Benign

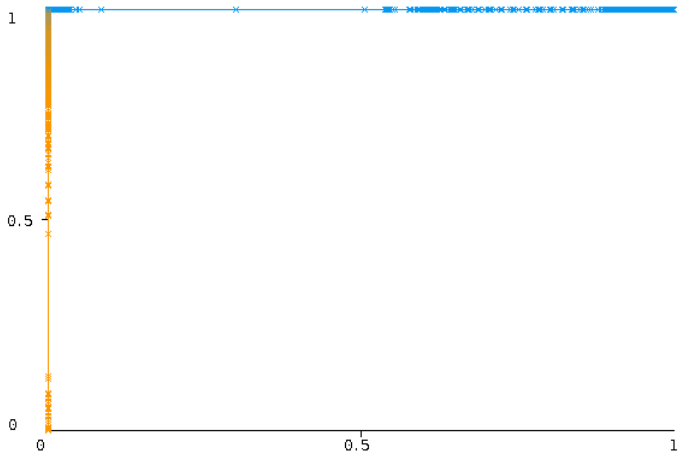

Figure A.2: CICIDS - 2017 Bot

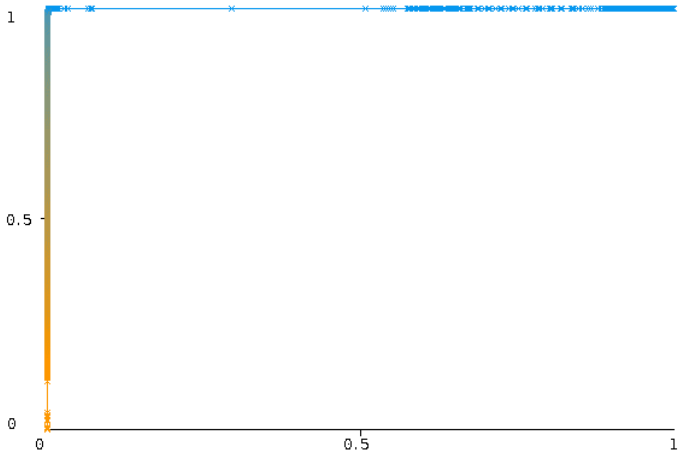

Figure A.3: CICIDS - 2017 BruteForce

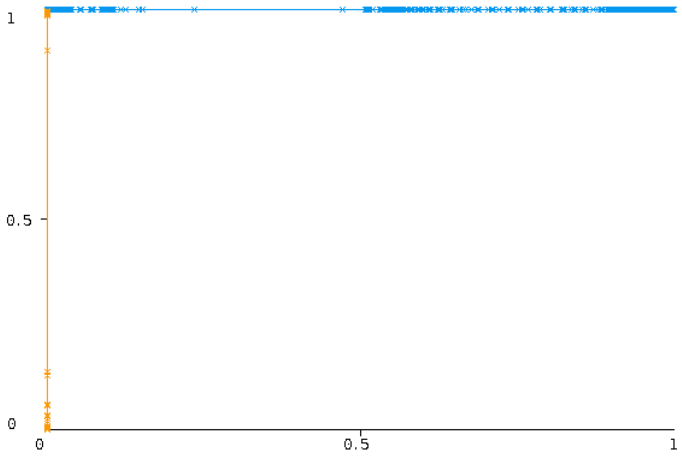

Figure A.4: CICIDS - 2017 DDoS

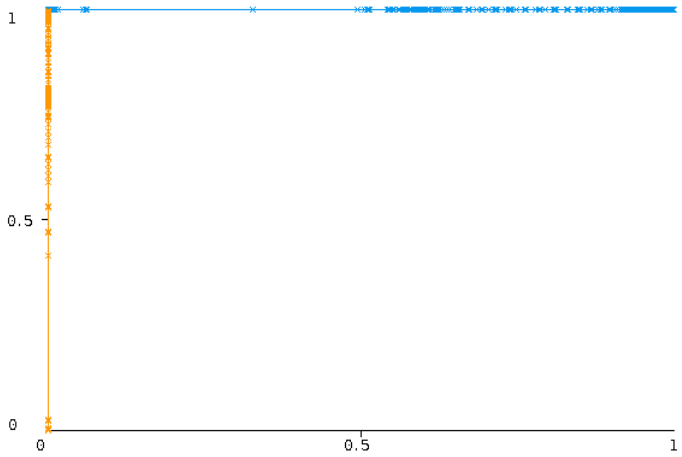

Figure A.5: CICIDS - 2017 Hulk

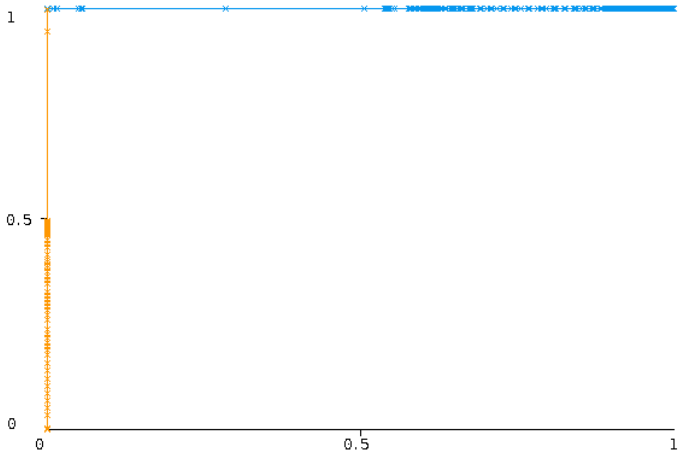

Figure A.6: CICIDS - 2017 FTP

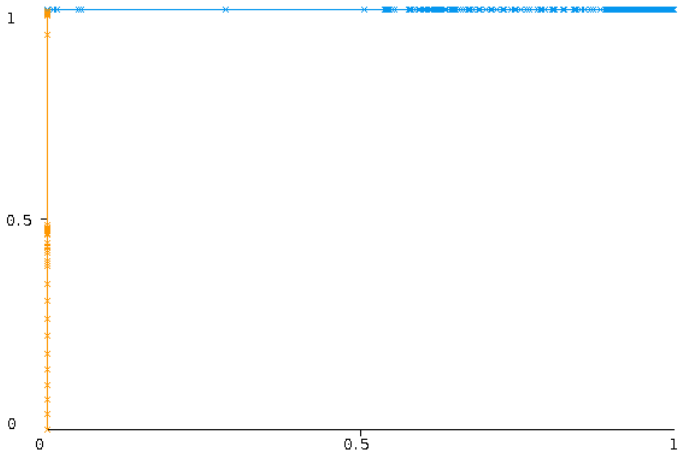

Figure A.7: CICIDS - 2017 SSH

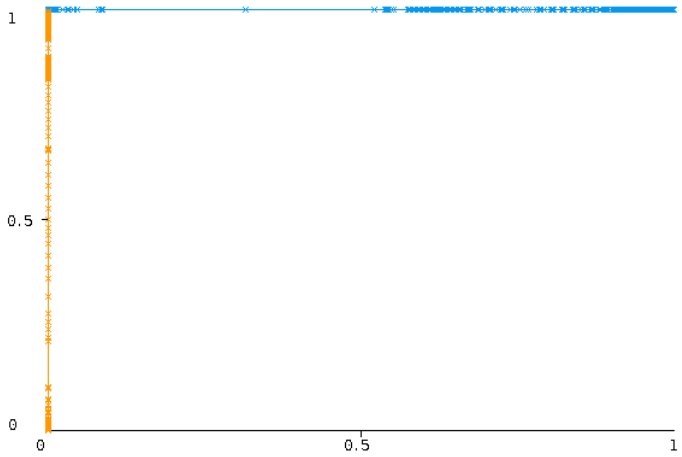

Figure A.8: CICIDS - 2017 GoldenEye

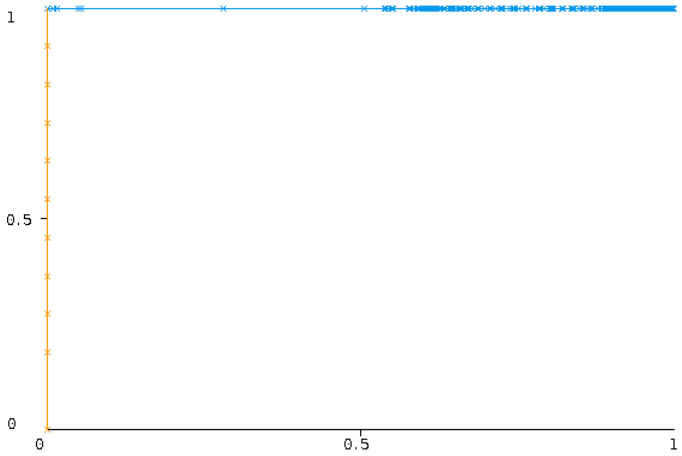

Figure A.9: CICIDS - 2017 HeartBleed

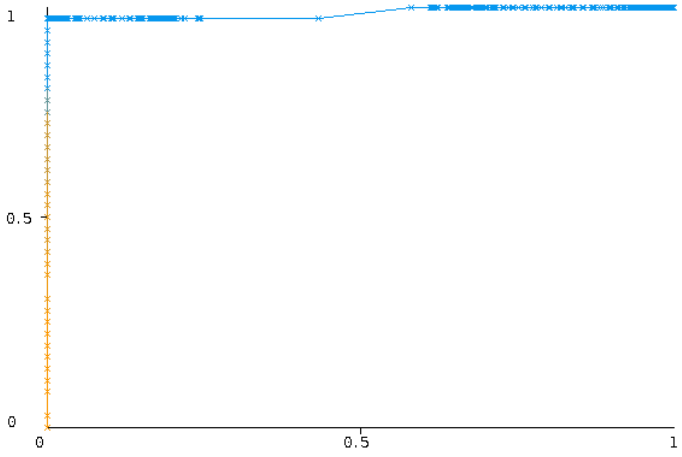

Figure A.10: CICIDS - 2017 Ilfiltration

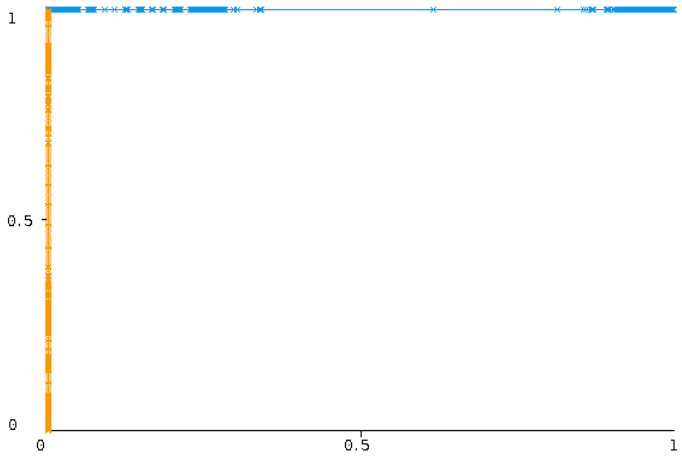

Figure A.11: CICIDS - 2017 Port Scan

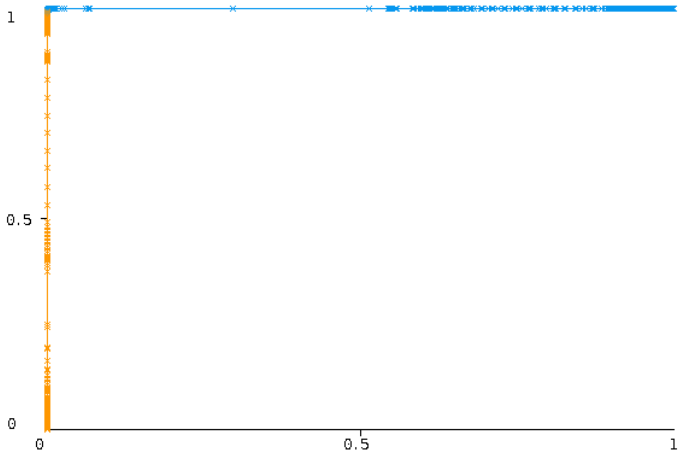

Figure A.12: CICIDS - 2017 SlowHTTP

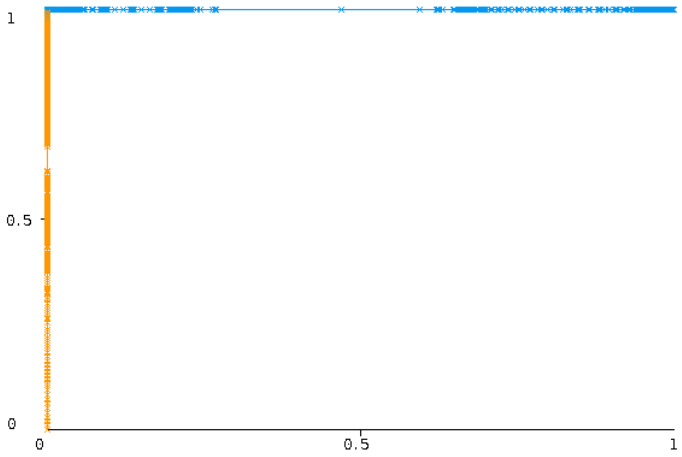

Figure A.13: CICIDS - 2017 Slowloris

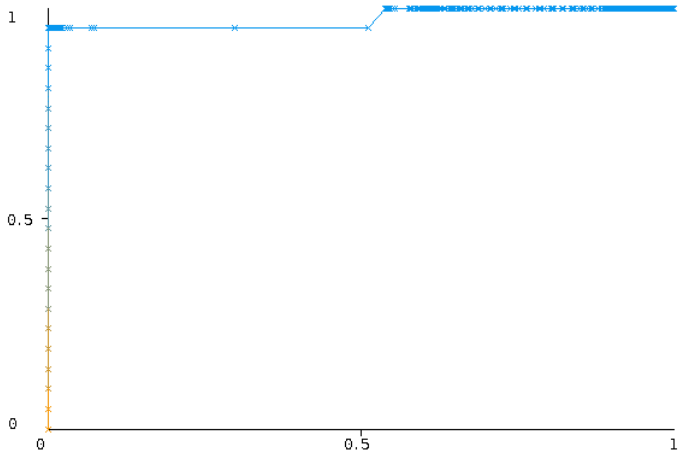

Figure A.14: CICIDS - 2017 SQL Injection

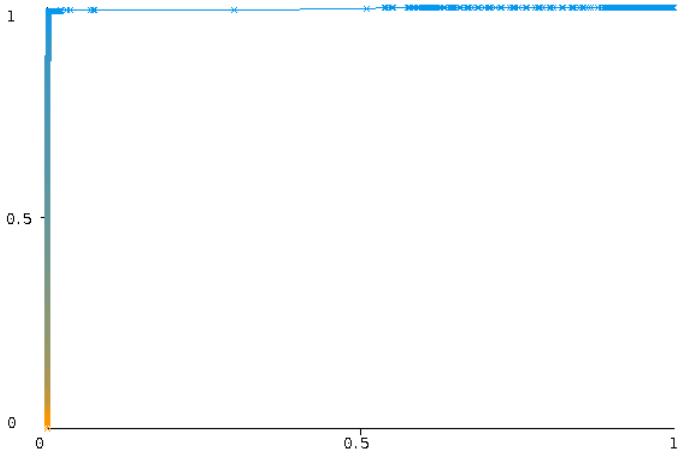

Figure A.15: CICIDS - 2017 XSS

The ROC values for our proposed ensemble-based feature selection and ensemble-based IDS framework using NSL - KDD have been listed in Table 5.11. The ROC graphs for Benign and each Attack class of the NSL - KDD dataset as shown in the Table 5.11 are represented in Figure A.16 to Figure A.20.

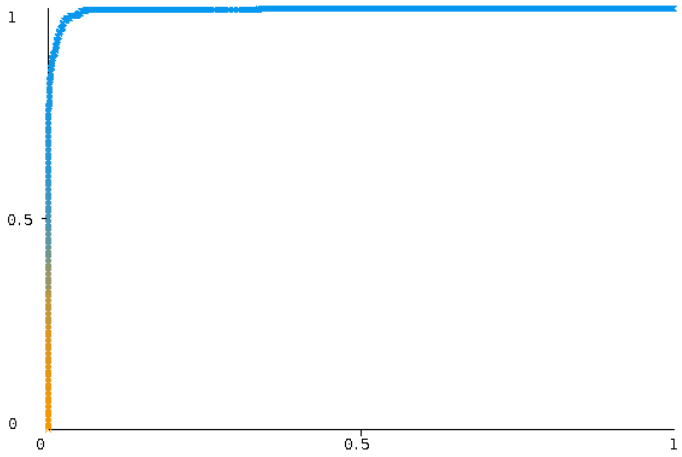

Figure A.16: NSL - KDD U2R

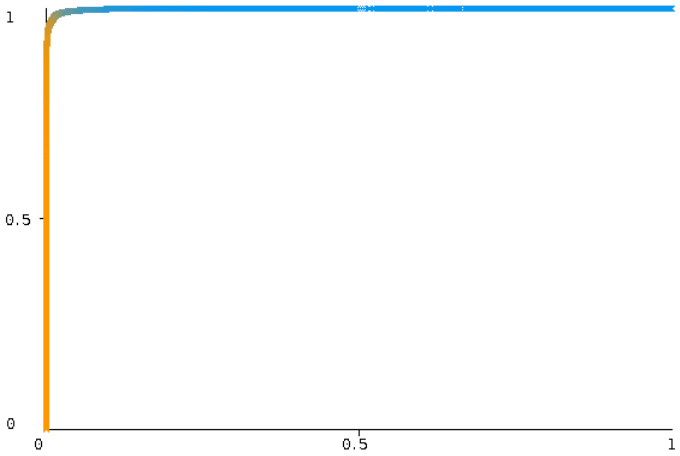

Figure A.17: NSL - KDD Normal

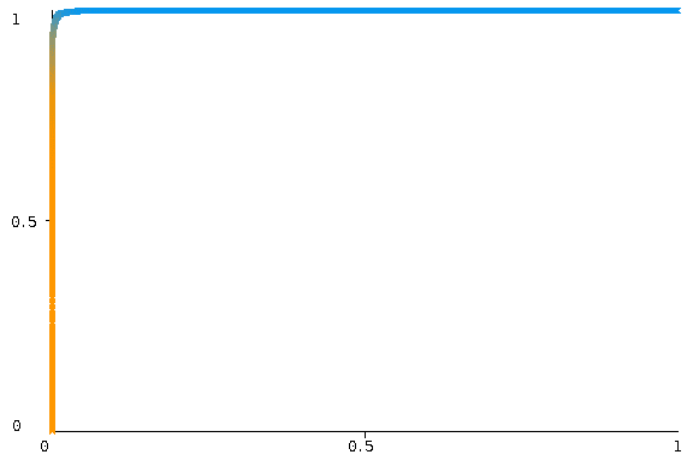

Figure A.18: NSL - KDD Probe

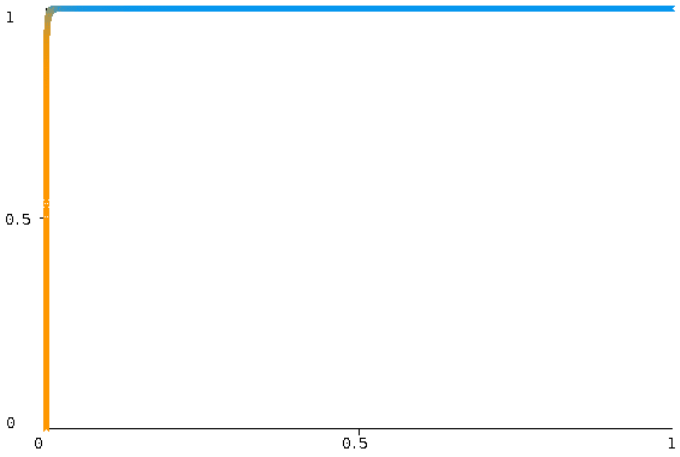

Figure A.19: NSL - KDD DDoS

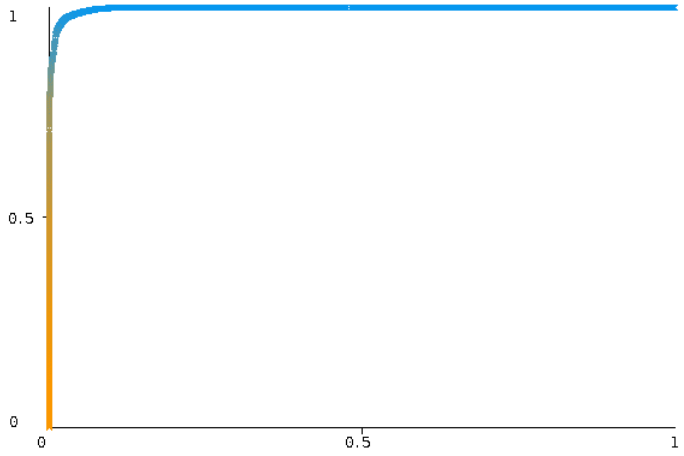

Figure A.20: NSL - KDD R2L

BIBLIOGRAPHY

[1] 451, *2021 technology preview a pivotal year for it in preparation for a postpandemic 'new normal'*.
https://www.spglobal.com/marketintelligence/en/documents/2021-technology-preview-a-pivotal.pdf.
Online; Accessed: 2020-11-09.

[2] M. ABIRAMI, U. YASH, AND S. SINGH, *Building an ensemble learning based algorithm for improving intrusion detection system*, in Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, 2020, pp. 635–649.

[3] A. A. ABUROMMAN AND M. B. I. REAZ, *A survey of intrusion detection systems based on ensemble and hybrid classifiers*, Computers & Security, 65 (2017), pp. 135–152.

[4] M. AHMED, A. NASER MAHMOOD, AND J. HU, *A survey of network anomaly detection techniques*, Journal of Network and Computer Applications, 60 (2016), pp. 19 – 31.

[5] S. ALJAWARNEH, M. B. YASSEIN, AND M. ALJUNDI, *An enhanced j48 classification algorithm for the anomaly intrusion detection systems*, Cluster Computing, 22 (2019), pp. 10549–10565.

[6] M. ALLOGHANI, D. AL-JUMEILY, J. MUSTAFINA, A. HUSSAIN, AND A. J. AL-JAAF, *A systematic review on supervised and unsupervised machine learning algorithms for data science*, Supervised and Unsupervised Learning for Data Science, (2020), pp. 3–21.

[7] M. A. AMBUSAIDI, X. HE, P. NANDA, AND Z. TAN, *Building an intrusion detection system using a filter-based feature selection algorithm*, IEEE transactions on computers, 65 (2016), pp. 2986–2998.

[8] M. A. AMBUSAIDI, X. HE, Z. TAN, P. NANDA, L. F. LU, AND U. T. NAGAR, *A novel feature selection approach for intrusion detection data classification*, in 2014 IEEE 13th international conference on trust, security and privacy in computing and communications, IEEE, 2014, pp. 82–89.

[9] P. AMINI, M. A. ARAGHIZADEH, AND R. AZMI, *A survey on botnet: Classification, detection and defense*, in 2015 International Electronics Symposium (IES), 2015, pp. 233–238.

[10] F. AMIRI, M. R. YOUSEFI, C. LUCAS, A. SHAKERY, AND N. YAZDANI, *Mutual information-based feature selection for intrusion detection systems*, Journal of Network and Computer Applications, 34 (2011), pp. 1184–1199.

[11] J. P. ANDERSON, *Computer security threat monitoring and surveillance*, Technical Report, James P. Anderson Company, (1980).

[12] K. ANUSHA AND E. SATHIYAMOORTHY, *Comparative study for feature selection algorithms in intrusion detection system*, Automatic Control and Computer Sciences, 50 (2016), pp. 1–9.

[13] H. ATTAK, M. COMBALIA, G. GARDIKIS, B. GASTÓN, L. JACQUIN, D. KATSIANIS, A. LITKE, N. PAPADAKIS, D. PAPADOPOULOS, A. PASTOR, ET AL., *Application of distributed computing and machine learning technologies to cybersecurity*, Space, 2 (2018), p. I2CAT.

[14] S. AXELSSON, *Intrusion detection systems: A survey and taxonomy*. `https://neuro.bstu.by/ai/To-dom/My_research/Paper-0-again/For-research/D-mining/Anomaly-D/Intrusion-detection/taxonomy.pdf`. Online; Accessed: 2021-01-18.

[15] R. BACE AND P. MELL, *Nist special publication on intrusion detection systems*, tech. rep., DTIC Document, 2001.

[16] E. BAHRI, N. HARBI, AND H. N. HUU, *Approach based ensemble methods for better and faster intrusion detection*, in Computational Intelligence in Security for Information Systems, Springer, 2011, pp. 17–24.

[17] A. BAKSHI AND Y. B. DUJODWALA, *Securing cloud from ddos attacks using intrusion detection system in virtual machine*, in Communication Software and

Networks, 2010. ICCSN'10. Second International Conference on, IEEE, 2010, pp. 260–264.

[18] S. M. H. BAMAKAN, B. AMIRI, M. MIRZABAGHERI, AND Y. SHI, *A new intrusion detection approach using pso based multiple criteria linear programming*, Procedia Computer Science, 55 (2015), pp. 231–237.

[19] S. M. H. BAMAKAN, H. WANG, T. YINGJIE, AND Y. SHI, *An effective intrusion detection framework based on mclp/svm optimized by time-varying chaos particle swarm optimization*, Neurocomputing, 199 (2016), pp. 90–102.

[20] A. BANSAL AND S. KAUR, *Extreme gradient boosting based tuning for classification in intrusion detection systems*, in International Conference on Advances in Computing and Data Sciences, Springer, 2018, pp. 372–380.

[21] A. BANSAL AND S. KAUR, *Data dimensionality reduction (ddr) scheme for intrusion detection system using ensemble and standalone classifiers*, in International Conference on Advances in Computing and Data Sciences, Springer, 2019, pp. 436–451.

[22] R. R. R. BARBOSA, R. SADRE, A. PRAS, AND R. VAN DE MEENT, *Simpleweb/university of twente traffic traces data repository*, Centre for Telematics and Information Technology, University of Twente, (2010).

[23] P. BARTLETT, Y. FREUND, W. S. LEE, AND R. E. SCHAPIRE, *Boosting the margin: A new explanation for the effectiveness of voting methods*, The annals of statistics, 26 (1998), pp. 1651–1686.

[24] R. BATTITI, *Using mutual information for selecting features in supervised neural net learning*, IEEE Transactions on neural networks, 5 (1994), pp. 537–550.

[25] M. H. BHUYAN, D. K. BHATTACHARYYA, AND J. K. KALITA, *Network anomaly detection: Methods, systems and tools*, IEEE Communications Surveys Tutorials, 16 (2014), pp. 303–336.

[26] A. BINBUSAYYIS AND T. VAIYAPURI, *Identifying and benchmarking key features for cyber intrusion detection: an ensemble approach*, IEEE Access, 7 (2019), pp. 106495–106513.

[27] A. BIVENS, C. PALAGIRI, R. SMITH, B. SZYMANSKI, M. EMBRECHTS, ET AL., *Network-based intrusion detection using neural networks*, Intelligent Engineering Systems through Artificial Neural Networks, 12 (2002), pp. 579–584.

[28] V. BOLON-CANEDO, N. SANCHEZ-MARONO, AND A. ALONSO-BETANZOS, *Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset*, Expert Systems with Applications, 38 (2011), pp. 5947–5957.

[29] E. BONABEAU, D. D. R. D. F. MARCO, M. DORIGO, G. THÉRAULAZ, G. THERAULAZ, ET AL., *Swarm intelligence: from natural to artificial systems*, no. 1, Oxford university press, 1999.

[30] A. BORJI, *Combining heterogeneous classifiers for network intrusion detection*, in Annual Asian Computing Science Conference, Springer, 2007, pp. 254–260.

[31] F. BOTES, L. LEENEN, AND R. DE LA HARPE, *Ant colony induced decision trees for intrusion detection*, in 16th European Conference on Cyber Warfare and Security, ACPI, 2017, pp. 53–62.

[32] F. BOTES, L. LEENEN, AND R. DE LA HARPE, *Ant colony induced decision trees for intrusion detection*, (2017).

[33] L. BREIMAN, *Bagging predictors*, Machine learning, 24 (1996), pp. 123–140.

[34] L. BREIMAN, *Random forests*, Machine learning, 45 (2001), pp. 5–32.

[35] A. L. BUCZAK AND E. GUVEN, *A survey of data mining and machine learning methods for cyber security intrusion detection*, IEEE Communications surveys & tutorials, 18 (2015), pp. 1153–1176.

[36] A. CHAN, W. W. NG, D. S. YEUNG, E. C. TSANG, ET AL., *Comparison of different fusion approaches for network intrusion detection using ensemble of rbfnn*, in Proceedings of 2005 international conference on machine learning and cybernetics, vol. 6, 2005, pp. 18–21.

[37] V. CHANDOLA, A. BANERJEE, AND V. KUMAR, *Anomaly detection: A survey*, ACM Comput. Surv., 41 (2009).

[38] G. CHANDRASHEKAR AND F. SAHIN, *A survey on feature selection methods*, Computers & Electrical Engineering, 40 (2014), pp. 16–28.

[39] S. CHEBROLU, A. ABRAHAM, AND J. P. THOMAS, *Hybrid feature selection for modeling intrusion detection systems*, in International Conference on Neural Information Processing, Springer, 2004, pp. 1020–1025.

[40] S. CHEBROLU, A. ABRAHAM, AND J. P. THOMAS, *Feature deduction and ensemble design of intrusion detection systems*, Computers & security, 24 (2005), pp. 295–307.

[41] Y. CHEN, Y. LI, X.-Q. CHENG, AND L. GUO, *Survey and taxonomy of feature selection algorithms in intrusion detection system*, Springer, 2006, pp. 153–167.

[42] D. CHOU AND M. JIANG, *Data-driven network intrusion detection: A taxonomy of challenges and methods*, arXiv preprint arXiv:2009.07352, (2020).

[43] Y. Y. CHUNG AND N. WAHID, *A hybrid network intrusion detection system using simplified swarm optimization (sso)*, Applied Soft Computing, 12 (2012), pp. 3014–3022.

[44] CISCO, *Global - 2021 forecast highlights*.
`https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf`.
Online; Accessed: 2020-11-09.

[45] CISCO, *What are the most common cyber attacks?*
`https://www.cisco.com/c/en_au/products/security/common-cyberattacks.html`.
Online; Accessed: 2020-10-18.

[46] J. CLEMENT, *Internet usage worldwide ‚Äì statistics & facts*.
`https://www.statista.com/topics/1145/internet-usage-worldwide/`.
Online; Accessed: 2020-10-18.

[47] C. CRANE, *Re-hash: The largest ddos attacks in history*.
`https://www.thesslstore.com/blog/largest-ddos-attack-in-history/`.
Online; Accessed: 2020-11-24.

[48] G. CREECH AND J. HU, *Generation of a new ids test dataset: Time to retire the kdd collection*, in 2013 IEEE Wireless Communications and Networking Conference (WCNC), 2013, pp. 4487–4492.

[49] Z. CUI, Y. CHANG, J. ZHANG, X. CAI, AND W. ZHANG, *Improved nsga-iii with selection-and-elimination operator*, Swarm and Evolutionary Computation, 49 (2019), pp. 23–33.

[50] P. CUNNINGHAM AND J. CARNEY, *Diversity versus quality in classification ensembles based on feature selection*, in European Conference on Machine Learning, Springer, 2000, pp. 109–116.

[51] R. K. CUNNINGHAM, R. P. LIPPMANN, D. J. FRIED, S. L. GARFINKEL, I. GRAF, K. R. KENDALL, S. E. WEBSTER, D. WYSCHOGROD, AND M. A. ZISSMAN, *Evaluating intrusion detection systems without attacking your friends: The 1998 darpa intrusion detection evaluation*, tech. rep., MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 1999.

[52] N. DAS, *The shellshock attack*.
`https://www.exploit-db.com/docs/48112`.
Online; Accessed: 2021-01-18.

[53] A. V. DASTJERDI, K. A. BAKAR, AND S. G. H. TABATABAEI, *Distributed intrusion detection in clouds using mobile agents*, in Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP'09. Third International Conference on, IEEE, 2009, pp. 175–180.

[54] H. DEBAR, M. DACIER, AND A. WESPI, *A revised taxonomy for intrusion-detection systems*, Annales Des Telecommunications, 55 (2000), pp. 361–378.

[55] D. DENNING AND P. G. NEUMANN, *Requirements and model for IDES-a real-time intrusion-detection expert system*, vol. 8, SRI International Menlo Park, 1985.

[56] T. G. DIETTERICH ET AL., *Ensemble learning*, The handbook of brain theory and neural networks, 2 (2002), pp. 110–125.

[57] S. DUA AND X. DU, *Data mining and machine learning in cybersecurity*, CRC press, 2016.

[58] R. EBERHART AND J. KENNEDY, *A new optimizer using particle swarm theory*, in MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Ieee, 1995, pp. 39–43.

[59] A. S. EESA, Z. ORMAN, AND A. M. A. BRIFCANI, *A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems*, Expert systems with applications, 42 (2015), pp. 2670–2679.

[60] S. ELHAG, A. FERNÁNDEZ, A. ALTALHI, S. ALSHOMRANI, AND F. HERRERA, *A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems*, Soft Computing, 23 (2019), pp. 1321–1336.

[61] F. B. ET AL., *Nslkdd-dataset*.
`https://github.com/InitRoot/NSLKDD-Dataset`.
Online; Accessed: 2021-03-21.

[62] M. A. FERRAG, L. MAGLARAS, S. MOSCHOYIANNIS, AND H. JANICKE, *Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study*, Journal of Information Security and Applications, 50 (2020), p. 102419.

[63] G. FOLINO, C. PIZZUTI, AND G. SPEZZANO, *An ensemble-based evolutionary framework for coping with distributed intrusion detection*, Genetic Programming and Evolvable Machines, 11 (2010), pp. 131–146.

[64] G. FOLINO AND P. SABATINO, *Ensemble based collaborative and distributed intrusion detection systems: A survey*, Journal of Network and Computer Applications, 66 (2016), pp. 1–16.

[65] Y. FREUND, *Boosting a weak learning algorithm by majority*, Information and computation, 121 (1995), pp. 256–285.

[66] J. FRUHLINGER, *What is a cyber attack? recent examples show disturbing trends*.
`https://www.csoonline.com/article/3237324/what-is-a-cyber-attack-recent-examples-show-disturbing-trends.html`.
Online; Accessed: 2020-11-24.

[67] N. GALOV, *25 must-know cloud computing statistics in 2020*.
`https://hostingtribunal.com/blog/cloud-computing-statistics/#gref`.
Online; Accessed: 2020-11-20.

[68] T. GARFINKEL, M. ROSENBLUM, ET AL., *A virtual machine introspection based architecture for intrusion detection.*, in Ndss, vol. 3, 2003, pp. 191–206.

[69] A. GHARIB, I. SHARAFALDIN, A. H. LASHKARI, AND A. A. GHORBANI, *An evaluation framework for intrusion detection dataset*, in 2016 International Conference on Information Science and Security (ICISS), 2016, pp. 1–6.

[70] P. GHOSH, A. KARMAKAR, J. SHARMA, AND S. PHADIKAR, *Cs-pso based intrusion detection system in cloud environment*, in Emerging Technologies in Data Mining and Information Security, A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, eds., Singapore, 2019, Springer Singapore, pp. 261–269.

[71] GOOGLE, *Machine learning crash course - classification : Roc and auc*. `https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc`.
Online; Accessed: 2021-03-21.

[72] M. G. GOUDA AND A. X. LIU, *Structured firewall design*, Computer Networks, 51 (2007), pp. 1106 – 1120.

[73] M. GOVINDARAJAN AND R. CHANDRASEKARAN, *Intrusion detection using an ensemble of classification methods*, in Proc. of the World Congress on Engineering and Computer Science, vol. 1, 2012, pp. 459–464.

[74] J. GU AND S. LU, *An effective intrusion detection approach using svm with naïve bayes feature embedding*, Computers & Security, (2020), p. 102158.

[75] Y. GUAN AND J. BAO, *A cp intrusion detection strategy on cloud computing*, in International Symposium on Web Information Systems and Applications (WISA), 2009, pp. 84–87.

[76] M. GUDADHE, P. PRASAD, AND L. K. WANKHADE, *A new data mining based network intrusion detection model*, in 2010 International Conference on Computer and Communication Technology (ICCCT), IEEE, 2010, pp. 731–735.

[77] W. HE, H. LI, AND J. LI, *Ensemble feature selection for improving intrusion detection classification accuracy*, in Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science, 2019, pp. 28–33.

[78] P. HICK, E. ABEN, K. CLAFFY, AND J. POLTEROCK, *the caida ddos attack 2007 dataset*, 2007.

[79] H. HINDY, D. BROSSET, E. BAYNE, A. SEEAM, C. TACHTATZIS, R. C. ATKINSON, AND X. J. A. BELLEKENS, *A taxonomy and survey of intrusion detection system design techniques, network threats and datasets*, CoRR, abs/1806.03517 (2018).

[80] T. K. HO, *The random subspace method for constructing decision forests*, IEEE transactions on pattern analysis and machine intelligence, 20 (1998), pp. 832–844.

[81] S.-J. HORNG, M.-Y. SU, Y.-H. CHEN, T.-W. KAO, R.-J. CHEN, J.-L. LAI, AND C. D. PERKASA, *A novel intrusion detection system based on hierarchical clustering and support vector machines*, Expert systems with Applications, 38 (2011), pp. 306–313.

[82] X. HU, *Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications*, in Proceedings 2001 IEEE International Conference on Data Mining, IEEE, 2001, pp. 233–240.

[83] J. HUANG, Y. CAI, AND X. XU, *A hybrid genetic algorithm for feature selection wrapper based on mutual information*, Pattern recognition letters, 28 (2007), pp. 1825–1844.

[84] D. HUBBARD, M. SUTTON, ET AL., *Top threats to cloud computing v1. 0*, Cloud Security Alliance, (2010).

[85] S. IQBAL, M. L. M. KIAH, B. DHAGHIGHI, M. HUSSAIN, S. KHAN, M. K. KHAN, AND K.-K. R. CHOO, *On cloud security attacks: A taxonomy and intrusion detection and prevention as a service*, Journal of Network and Computer Applications, 74 (2016), pp. 98–120.

[86] S. JIANG AND X. XU, *Application and performance analysis of data preprocessing for intrusion detection system*, in International Conference on Science of Cyber Security, Springer, 2019, pp. 163–177.

[87] V. KANIMOZHI AND T. P. JACOB, *Calibration of various optimized machine learning classifiers in network intrusion detection system on the realistic cyber dataset cse-cic-ids2018 using cloud computing*, International Journal of Engineering Applied Sciences and Technology, 4 (2019), pp. 2455–2143.

[88] C. KHAMMASSI AND S. KRICHEN, *A ga-lr wrapper approach for feature selection in network intrusion detection*, computers & security, 70 (2017), pp. 255–277.

[89] M. T. KHORSHED, A. S. ALI, AND S. A. WASIMI, *A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing*, Future Generation computer systems, 28 (2012), pp. 833–851.

[90] A. KHRAISAT, I. GONDAL, P. VAMPLEW, AND J. KAMRUZZAMAN, *Survey of intrusion detection systems: techniques, datasets and challenges*, Cybersecurity, 2 (2019), p. 20.

[91] L. I. KUNCHEVA, *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley and Sons, Inc., 2004.

[92] L. I. KUNCHEVA, *Combining pattern classifiers: methods and algorithms*, John Wiley & Sons, 2014.

[93] N. KWAK AND C.-H. CHOI, *Input feature selection for classification problems*, IEEE transactions on neural networks, 13 (2002), pp. 143–159.

[94] H. LASHKARI, *Cicflowmeter*.
`https://www.unb.ca/cic/research/applications.html`.
Online; Accessed: 2021-03-15.

[95] A. LAZAREVIC, V. KUMAR, AND J. SRIVASTAVA, *Intrusion Detection: A Survey*, Springer US, Boston, MA, 2005, pp. 19–78.

[96] W. LEE AND S. J. STOLFO, *A framework for constructing features and models for intrusion detection systems*, ACM Trans. Inf. Syst. Secur., 3 (2000), p. 227,Äì261.

[97] J. LI, K. CHENG, S. WANG, F. MORSTATTER, R. P. TREVINO, J. TANG, AND H. LIU, *Feature selection: A data perspective*, ACM Computing Surveys (CSUR), 50 (2017), pp. 1–45.

[98] H.-J. LIAO, C.-H. RICHARD LIN, Y.-C. LIN, AND K.-Y. TUNG, *Intrusion detection system: A comprehensive review*, Journal of Network and Computer Applications, 36 (2013), pp. 16 – 24.

[99] M. LIBERATORE AND P. SHENOY, *Umass trace repository*, Accessed: May, (2017).

[100] L. LIN, R. ZUO, S. YANG, AND Z. ZHANG, *Svm ensemble for anomaly detection based on rotation forest*, in 2012 Third International Conference on Intelligent Control and Information Processing, IEEE, 2012, pp. 150–153.

[101] M. LIU, Z. XUE, X. XU, C. ZHONG, AND J. CHEN, *Host-based intrusion detection system with system calls: Review and future trends*, ACM Computing Surveys (CSUR), 51 (2018), pp. 1–36.

[102] C.-C. LO, C.-C. HUANG, AND J. KU, *A cooperative intrusion detection system framework for cloud computing networks*, in Parallel processing workshops (ICPPW), 2010 39th international conference on, IEEE, 2010, pp. 280–284.

[103] G. MAAYAN, *The iot rundown for 2020: Stats, risks, and solutions*.
https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx#:~:text=Every%20second%E2%80%94127%20new%20IoT,be%20connected%20to%20the%20web.
Online; Accessed: 2020-11-20.

[104] G. D. MAAYAN, *The iot rundown for 2020: Stats, risks, and solutions*.
https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx?
Online; Accessed: 2021-03-21.

[105] A. J. MALIK, W. SHAHZAD, AND F. A. KHAN, *Binary pso and random forests algorithm for probe attacks detection in a network*, in 2011 IEEE Congress of Evolutionary Computation (CEC), IEEE, 2011, pp. 662–668.

[106] I. MANZOOR, N. KUMAR, ET AL., *A feature reduced intrusion detection system using ann classifier*, Expert Systems with Applications, 88 (2017), pp. 249–257.

[107] C. MAZZARIELLO, R. BIFULCO, AND R. CANONICO, *Integrating a network ids into an open source cloud computing environment*, in Information Assurance and Security (IAS), 2010 Sixth International Conference on, IEEE, 2010, pp. 265–270.

[108] J. MCHUGH, *Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory*, ACM Transactions on Information and System Security (TISSEC), 3 (2000), pp. 262–294.

[109] T. MEHMOD AND H. B. M. RAIS, *Ant colony optimization and feature selection for intrusion detection*, in Advances in machine learning and signal processing, Springer, 2016, pp. 305–312.

[110] P. MELL, T. GRANCE, ET AL., *The nist definition of cloud computing*, (2011).

[111] P. MISHRA, V. VARADHARAJAN, U. TUPAKULA, AND E. S. PILLI, *A detailed investigation and analysis of using machine learning techniques for intrusion detection*, IEEE Communications Surveys Tutorials, 21 (2019), pp. 686–728.

[112] C. MODI, D. PATEL, B. BORISANIYA, H. PATEL, A. PATEL, AND M. RAJARAJAN, *A survey of intrusion detection techniques in cloud*, Journal of Network and Computer Applications, 36 (2013), pp. 42 – 57.

[113] C. MODI, D. PATEL, B. BORISANIYA, H. PATEL, A. PATEL, AND M. RAJARAJAN, *A survey of intrusion detection techniques in cloud*, Journal of Network and Computer Applications, 36 (2013), pp. 42–57.

[114] C. N. MODI AND K. ACHA, *Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review*, the Journal of Supercomputing, 73 (2017), pp. 1192–1234.

[115] N. MOUSTAFA, *The unsw - nb15 dataset description*.
`https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/`.
Online; Accessed: 2021-03-15.

[116] N. MOUSTAFA, J. HU, AND J. SLAY, *A holistic review of network anomaly detection systems: A comprehensive survey*, Journal of Network and Computer Applications, 128 (2019), pp. 33 – 55.

[117] N. MOUSTAFA AND J. SLAY, *Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)*, in 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6.

[118] N. MOUSTAFA AND J. SLAY, *Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)*, in 2015 military communications and information systems conference (MilCIS), IEEE, 2015, pp. 1–6.

[119] N. MOUSTAFA AND J. SLAY, *A hybrid feature selection for network intrusion detection systems: Central points*, arXiv preprint arXiv:1707.05505, (2017).

[120] S. MUKKAMALA AND A. H. SUNG, *Significant feature selection using computational intelligent techniques for intrusion detection*, in Advanced Methods for Knowledge Discovery from Complex Data, Springer, 2005, pp. 285–306.

[121] S. MUKKAMALA, A. H. SUNG, AND A. ABRAHAM, *Intrusion detection using an ensemble of intelligent paradigms*, Journal of network and computer applications, 28 (2005), pp. 167–182.

[122] U. NAGAR, P. NANDA, X. HE, AND Z. TAN, *A framework for data security in cloud using collaborative intrusion detection scheme*, in Proceedings of the 10th International Conference on Security of Information and Networks, 2017, pp. 188–193.

[123] P. NANDA, A. ARAIN, AND U. NAGAR, *Network packet breach detection using cognitive techniques*, in Smart Systems and IoT: Innovations in Computing, Springer, 2020, pp. 555–565.

[124] B. NECHAEV, M. ALLMAN, V. PAXSON, AND A. GURTOV, *Lawrence berkeley national laboratory (lbnl)/icsi enterprise tracing project*, Berkeley, CA: LBNL/ICSI, (2004).

[125] P. NEVAVUORI AND T. KOKKONEN, *Requirements for training and evaluation dataset of network and host intrusion detection system*, in New Knowledge in Information Systems and Technologies, Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, eds., Cham, 2019, Springer International Publishing, pp. 534–546.

[126] B. H. NGUYEN, B. XUE, AND M. ZHANG, *A survey on swarm intelligence approaches to feature selection in data mining*, Swarm and Evolutionary Computation, 54 (2020), p. 100663.

[127] H. H. NGUYEN, N. HARBI, AND J. DARMONT, *An efficient local region and clustering-based ensemble system for intrusion detection*, in Proceedings of the 15th Symposium on International Database Engineering & Applications, 2011, pp. 185–191.

[128] P. NICHOLSON, *Five most famous ddos attacks and then some*. `https://www.a10networks.com/blog/5-most-famous-ddos-attacks/`. Online; Accessed: 2020-11-24.

[129] L. S. OLIVEIRA, R. SABOURIN, F. BORTOLOZZI, AND C. Y. SUEN, *A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition*, International Journal of Pattern Recognition and Artificial Intelligence, 17 (2003), pp. 903–929.

[130] O. OSANAIYE, H. CAI, K.-K. R. CHOO, A. DEHGHANTANHA, Z. XU, AND M. DLODLO, *Ensemble-based multi-filter feature selection method for ddos detection in cloud computing*, EURASIP Journal on Wireless Communications and Networking, 2016 (2016), pp. 1–10.

[131] OSSEC.
https://www.ossec.net/.
Online; Accessed: 2021-01-18.

[132] S. M. OTHMAN, F. M. BA-ALWI, N. T. ALSOHYBE, AND A. Y. AL-HASHIDA, *Intrusion detection model using machine learning algorithm on big data environment*, Journal of Big Data, 5 (2018), pp. 1–12.

[133] J. S. PARK, K. M. SHAZZAD, AND D. S. KIM, *Toward modeling lightweight intrusion detection system through correlation-based hybrid feature selection*, in International Conference on Information Security and Cryptology, Springer, 2005, pp. 279–289.

[134] A. PENN, *Australia's 2020 cyber security strategy*.
https://www.homeaffairs.gov.au/cyber-security-subsite/files/2020-cyber-security-strategy-iap-report.pdf.
Online; Accessed: 2020-10-18.

[135] R. PERDISCI, D. ARIU, P. FOGLA, G. GIACINTO, AND W. LEE, *Mcpad: A multiple classifier system for accurate payload-based anomaly detection*, Computer networks, 53 (2009), pp. 864–881.

[136] R. PERDISCI, G. GU, AND W. LEE, *Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems*, in Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 488–498.

[137] C. PETTEY, *Gartner top 9 security and risk trends for 2020*.
https://www.gartner.com/smarterwithgartner/gartner-top-9-security-and-risk-trends-for-2020/.
Online; Accessed: 2020-10-18.

[138] J. R. QUINLAN, *C4. 5: programs for machine learning*, Elsevier, 2014.

[139] H. RAJADURAI AND U. D. GANDHI, *A stacked ensemble learning model for intrusion detection in wireless network*, Neural Computing and Applications, (2020), pp. 1–9.

[140] P. A. A. RESENDE AND A. C. DRUMMOND, *A survey of random forest based methods for intrusion detection systems*, ACM Computing Surveys (CSUR), 51 (2018), pp. 1–36.

[141] M. RING, S. WUNDERLICH, D. SCHEURING, D. LANDES, AND A. HOTHO, *A survey of network-based intrusion detection data sets*, Computers & Security, 86 (2019), pp. 147–167.

[142] L. ROKACH, B. CHIZI, AND O. MAIMON, *Feature selection by combining multiple methods*, in Advances in Web Intelligence and Data Mining, Springer, 2006, pp. 295–304.

[143] R. ROOHPARVAR, *What is a backdoor attack?* `https://www.infoguardsecurity.com/what-is-a-backdoor-attack/#:~:text=The%20backdoor%20attack%20is%20a,plug%2Dins%20or%20input%20fields`.
Online; Accessed: 2021-01-18.

[144] S. ROSCHKE, F. CHENG, AND C. MEINEL, *An extensible and virtualization-compatible ids management architecture*, in Information Assurance and Security, 2009. IAS'09. Fifth International Conference on, vol. 2, IEEE, 2009, pp. 130–134.

[145] S. ROSCHKE, F. CHENG, AND C. MEINEL, *Intrusion detection in the cloud*, in 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009, pp. 729–734.

[146] L. RUTKOWSKI, M. JAWORSKI, L. PIETRUCZUK, AND P. DUDA, *Decision trees for mining data streams based on the gaussian approximation*, IEEE Transactions on Knowledge and Data Engineering, 26 (2013), pp. 108–119.

[147] F. SALO, M. INJADAT, A. B. NASSIF, A. SHAMI, AND A. ESSEX, *Data mining techniques in intrusion detection systems: A systematic literature review*, IEEE Access, 6 (2018), pp. 56046–56058.

[148] B. SANGSTER, T. O'CONNOR, T. COOK, R. FANELLI, E. DEAN, C. MORRELL, AND G. J. CONTI, *Toward instrumenting network warfare competitions to generate labeled datasets.*, in CSET, 2009.

[149] T. SARANYA, S. SRIDEVI, C. DEISY, T. D. CHUNG, AND M. A. KHAN, *Performance analysis of machine learning algorithms in intrusion detection system: A review*, Procedia Computer Science, 171 (2020), pp. 1251–1260.

[150] B. SELVAKUMAR AND K. MUNEESWARAN, *Firefly algorithm based feature selection for network intrusion detection*, Computers & Security, 81 (2019), pp. 148–155.

[151] I. SHARAFALDIN, A. GHARIB, A. H. LASHKARI, AND A. A. GHORBANI, *Towards a reliable intrusion detection benchmark dataset*, Software Networking, 2018 (2018), pp. 177–200.

[152] I. SHARAFALDIN, A. HABIBI LASHKARI, AND A. A. GHORBANI, *A detailed analysis of the cicids2017 data set*, in Information Systems Security and Privacy, P. Mori, S. Furnell, and O. Camp, eds., Cham, 2019, Springer International Publishing, pp. 172–188.

[153] I. SHARAFALDIN, A. H. LASHKARI, AND A. A. GHORBANI, *Toward generating a new intrusion detection dataset and intrusion traffic characterization.*, in ICISSp, 2018, pp. 108–116.

[154] Z. SHI, J. LI, C. WU, AND J. LI, *Deepwindow: An efficient method for online network traffic anomaly detection*, in 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2019, pp. 2403–2408.

[155] A. SHIRAVI, H. SHIRAVI, M. TAVALLAEE, AND A. A. GHORBANI, *Toward developing a systematic approach to generate benchmark datasets for intrusion detection*, computers & security, 31 (2012), pp. 357–374.

[156] M. SIRAST, *What is confusion matrix and advance classification metrics?* `https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html`. Online; Accessed: 2021-03-21.

[157] SNORT. `https://www.snort.org/`.

Online; Accessed: 2021-01-18.

[158] R. SOMMER AND V. PAXSON, *Outside the closed world: On using machine learning for network intrusion detection*, in 2010 IEEE symposium on security and privacy, IEEE, 2010, pp. 305–316.

[159] J. SONG, H. TAKAKURA, Y. OKABE, M. ETO, D. INOUE, AND K. NAKAO, *Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation*, in Proceedings of the first workshop on building analysis datasets and gathering experience returns for security, 2011, pp. 29–36.

[160] J. SONG, W. ZHAO, Q. LIU, AND X. WANG, *Hybrid feature selection for supporting lightweight intrusion detection systems*, in Journal of Physics: Conference Series, vol. 887, IOP Publishing, 2017, p. 012031.

[161] B. STACKPOLE, *Symantec security summary-june 2020*.
https://symantec-enterprise-blogs.security.com/blogs/feature-stories/symantec-security-summary-june-2020.
Online; Accessed: 2020-10-18.

[162] STATISTA, *Number of consumer cloud-based service users worldwide in 2013 and 2018*.
https://www.statista.com/statistics/321215/global-consumer-cloud-computing-users/#:~:text=This%20statistic%20presents%20the%20number,2.4%20billion%20users%20in%202013.
Online; Accessed: 2020-11-20.

[163] D. STIAWAN, M. Y. B. IDRIS, A. M. BAMHDI, R. BUDIARTO, ET AL., *Cicids-2017 dataset feature analysis with information gain for anomaly detection*, IEEE Access, 8 (2020), pp. 132911–132921.

[164] J. STOLTZFUS, *What,Äôs the difference between sem, sim and siem?*
https://www.techopedia.com/7/31201/security/whats-the-difference-between-sem-sim-and-siem.
Online; Accessed: 2021-03-21.

[165] A. SUBAIRA AND P. ANITHA, *Efficient classification mechanism for network intrusion detection system based on data mining techniques: a survey*, in 2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO), IEEE, 2014, pp. 274–280.

[166] S. SUBASHINI AND V. KAVITHA, *A survey on security issues in service delivery models of cloud computing*, Journal of network and computer applications, 34 (2011), pp. 1–11.

[167] T. H. T. SYMANTEC, *Threat landscape trends-q1 2020*.
https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/threat-landscape-q1-2020.
Online; Accessed: 2020-11-24.

[168] SYMANTECA, *Internet security threat repost 2019*.
https://docs.broadcom.com/doc/istr-24-2019-en.
Online; Accessed: 2020-11-24.

[169] SYMANTECB, *2016 internet security threat report*.
https://docs.broadcom.com/doc/istr-21-2016-en.
Online; Accessed: 2020-11-24.

[170] B. A. TAMA AND S. LIM, *Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation*, Computer Science Review, 39 (2021), p. 100357.

[171] B. A. TAMA, L. NKENYEREYE, S. R. ISLAM, AND K.-S. KWAK, *An enhanced anomaly detection in web traffic using a stack of classifier ensemble*, IEEE Access, 8 (2020), pp. 24120–24134.

[172] B. A. TAMA AND K. H. RHEE, *A combination of pso-based feature selection and tree-based classifiers ensemble for intrusion detection systems*, in Advances in Computer Science and Ubiquitous Computing, Springer, 2015, pp. 489–495.

[173] Z. TAN, U. T. NAGAR, X. HE, P. NANDA, R. P. LIU, S. WANG, AND J. HU, *Enhancing big data security with collaborative intrusion detection*, IEEE cloud computing, 1 (2014), pp. 27–33.

[174] M. TAVALLAEE, E. BAGHERI, W. LU, AND A. A. GHORBANI, *A detailed analysis of the kdd cup 99 data set*, in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.

[175] M. TAVALLAEE, N. STAKHANOVA, AND A. A. GHORBANI, *Toward credible evaluation of anomaly-based intrusion-detection methods*, IEEE Transactions on

Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40 (2010), pp. 516–524.

[176] A. THAKKAR AND R. LOHIYA, *A review of the advancement in intrusion detection datasets*, Procedia Computer Science, 167 (2020), pp. 636–645.

[177] A. THAKKAR AND R. LOHIYA, *Attack classification using feature selection techniques: a comparative study*, Journal of Ambient Intelligence and Humanized Computing, 12 (2021), pp. 1249–1266.

[178] V. THAWARE, *Covid-19 outbreak prompts opportunistic wave of malicious email campaigns*.
https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/covid-19-outbreak-prompts-opportunistic-wave-malicious-email-campaigns.
Online; Accessed: 2020-11-24.

[179] A. TSYMBAL, S. PUURONEN, AND D. W. PATTERSON, *Ensemble feature selection with the simple bayesian classification*, Information fusion, 4 (2003), pp. 87–100.

[180] E. TUV AND K. TORKKOLA, *Feature filtering with ensembles using artificial contrasts*, Feature Selection for Data Mining, (2005), p. 69.

[181] UPASANA, *Real world iot applications in different domains*.
https://www.edureka.co/blog/iot-applications/.
Online; Accessed: 2020-11-20.

[182] P. R. K. VARMA, V. V. KUMARI, AND S. S. KUMAR, *A survey of feature selection techniques in intrusion detection system: A soft computing perspective*, in Progress in computing, analytics and networking, Springer, 2018, pp. 785–793.

[183] E. VASILOMANOLAKIS, S. KARUPPAYAH, M. MÜHLHÄUSER, AND M. FISCHER, *Taxonomy and survey of collaborative intrusion detection*, ACM Comput. Surv., 47 (2015).

[184] V. VEMURI, *Enhancing Computer Security with Smart Technology*, CRC Press, 2005.

[185] VERIZON, *Data breach inverstigations report 2020*.
https://dd80b675424c132b90b3-e48385e382d2e5d17821a5e1d8e4c86b.ssl.cf1.rackcdn.com/external/2020-verizon-data-breach-investigations-report.pdf.
Online; Accessed: 2020-10-18.

[186] K. VIEIRA, A. SCHULTER, C. WESTPHALL, AND C. WESTPHALL, *Intrusion detection techniques in grid and cloud computing environment*, IT Professional, IEEE Computer Society, 12 (2010), pp. 38–43.

[187] R. VIJAYANAND, D. DEVARAJ, AND B. KANNAPIRAN, *Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection*, Computers & Security, 77 (2018), pp. 304–314.

[188] R. VINAYAKUMAR, M. ALAZAB, K. P. SOMAN, P. POORNACHANDRAN, A. AL-NEMRAT, AND S. VENKATRAMAN, *Deep learning approach for intelligent intrusion detection system*, IEEE Access, 7 (2019), pp. 41525–41550.

[189] R. VINAYAKUMAR, K. SOMAN, AND P. POORNACHANDRAN, *Evaluating effectiveness of shallow and deep networks to intrusion detection system*, in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 1282–1289.

[190] M. ZAMANI AND M. MOVAHEDI, *Machine learning techniques for intrusion detection*, arXiv preprint arXiv:1312.2177, (2013).

[191] F. ZHAO, J. ZHAO, X. NIU, S. LUO, AND Y. XIN, *A filter feature selection algorithm based on mutual information for intrusion detection*, Applied Sciences, 8 (2018), p. 1535.

[192] Y. ZHONG, W. CHEN, Z. WANG, Y. CHEN, K. WANG, Y. LI, X. YIN, X. SHI, J. YANG, AND K. LI, *Helad: A novel network anomaly detection model based on heterogeneous ensemble learning*, Computer Networks, 169 (2020), p. 107049.

[193] Y. ZHOU, G. CHENG, S. JIANG, AND M. DAI, *Building an efficient intrusion detection system based on feature selection and ensemble classifier*, Computer Networks, 174 (2020), p. 107247.

[194] Z.-H. ZHOU, *Ensemble methods: foundations and algorithms*, CRC press, 2012.